National Czech Programme | eosc

git GitHub

Git & GitHub v kostce:

Praktický úvod do verzování pro výzkumníky

Marek Pačes

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

National Czech Programme | eosc

git GitHub

Program

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Program

1) Úvod
2) **Git**
3) **GitHub**
4) Q&A

National Czech Programme | eosc

git GitHub

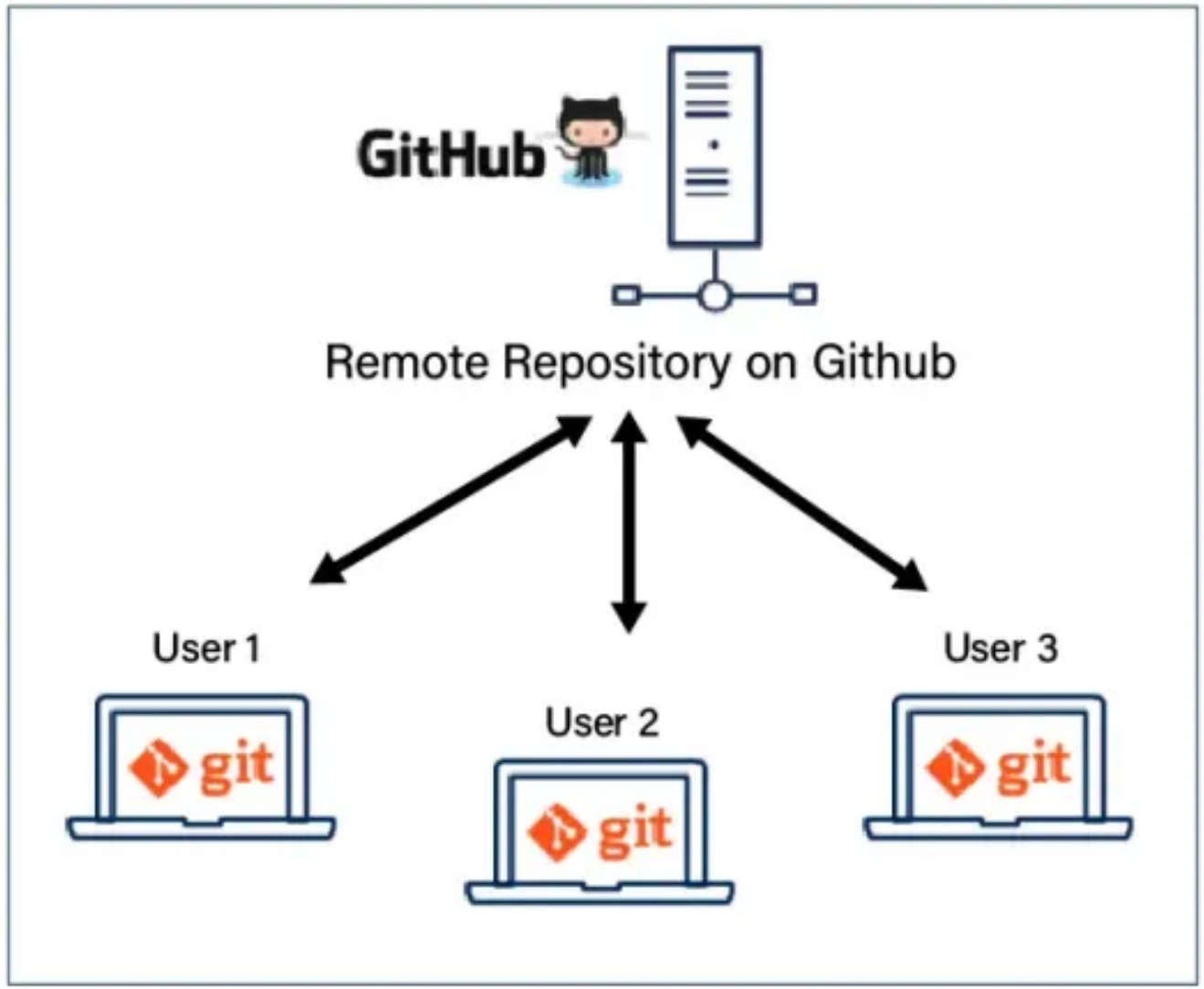
Úvod

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Úvod

- Verzování = Git
- Spolupráce = Git + GitHub
- Řízení projektu = GitHub



https://www.sprintzeal.com/blog/how-to-use-github

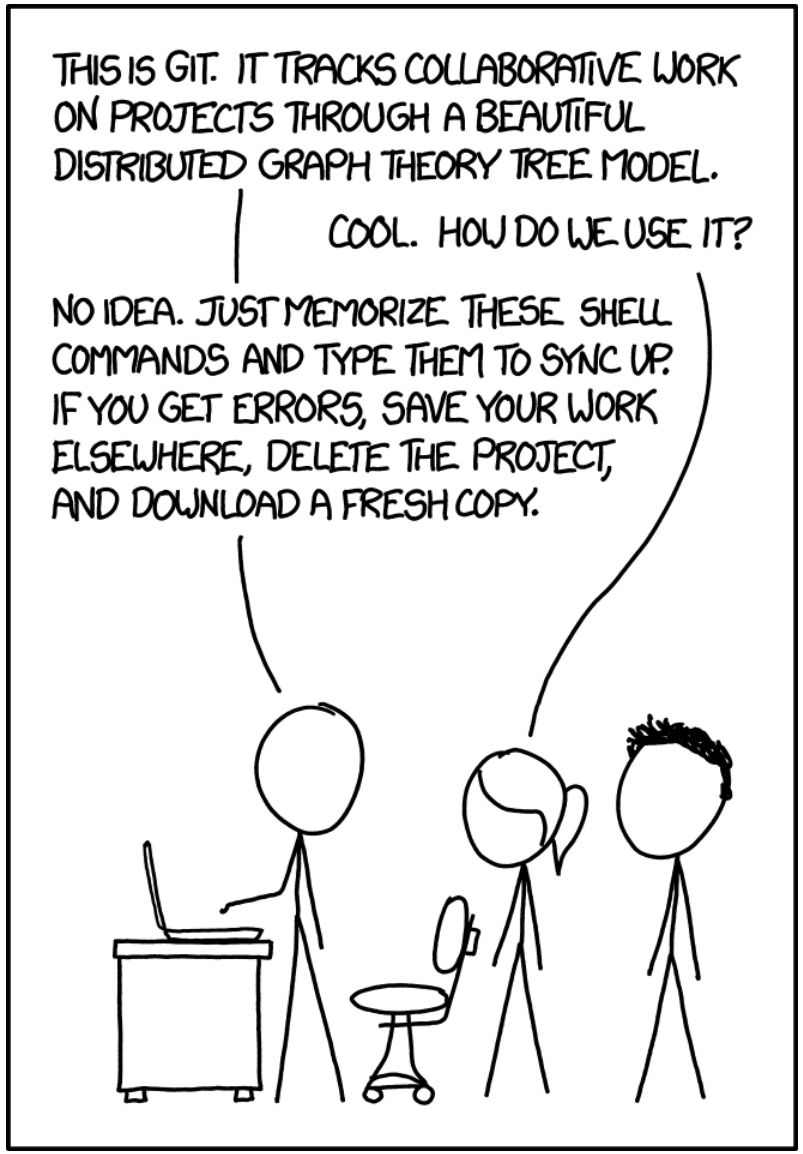
National Czech Programme | eosc

git

Git

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

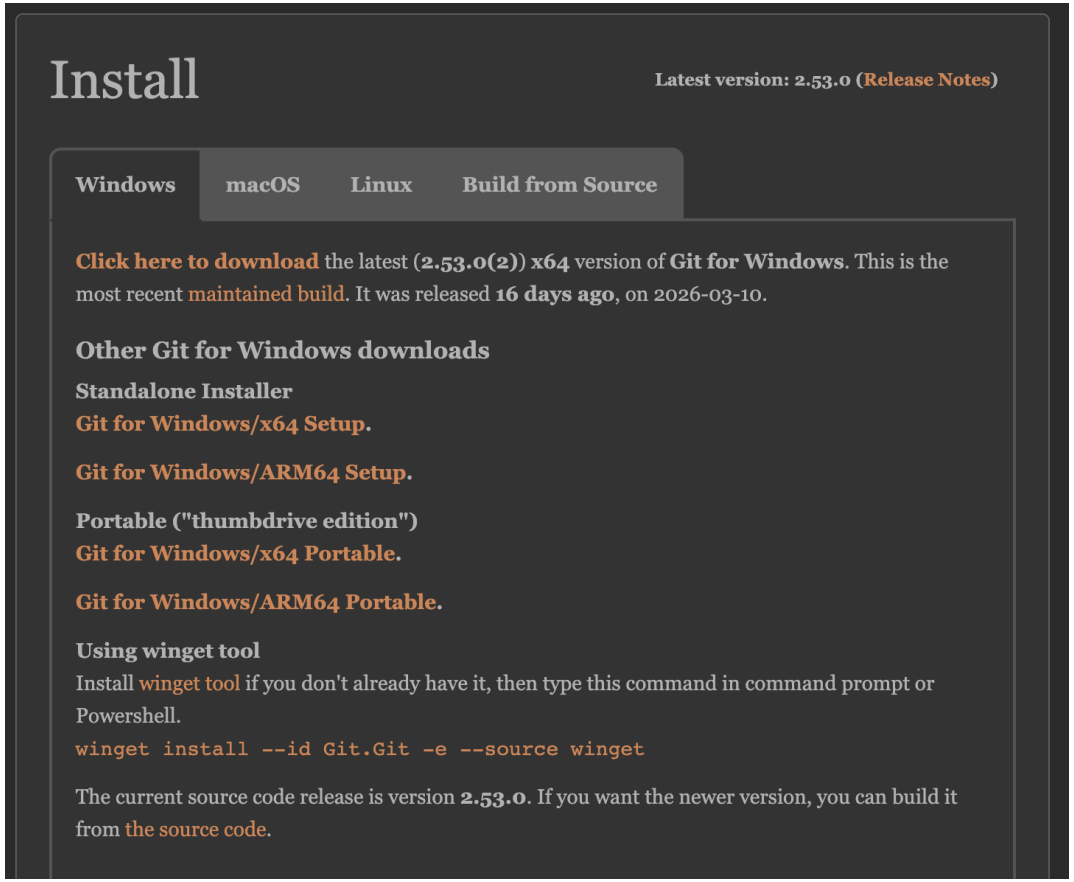


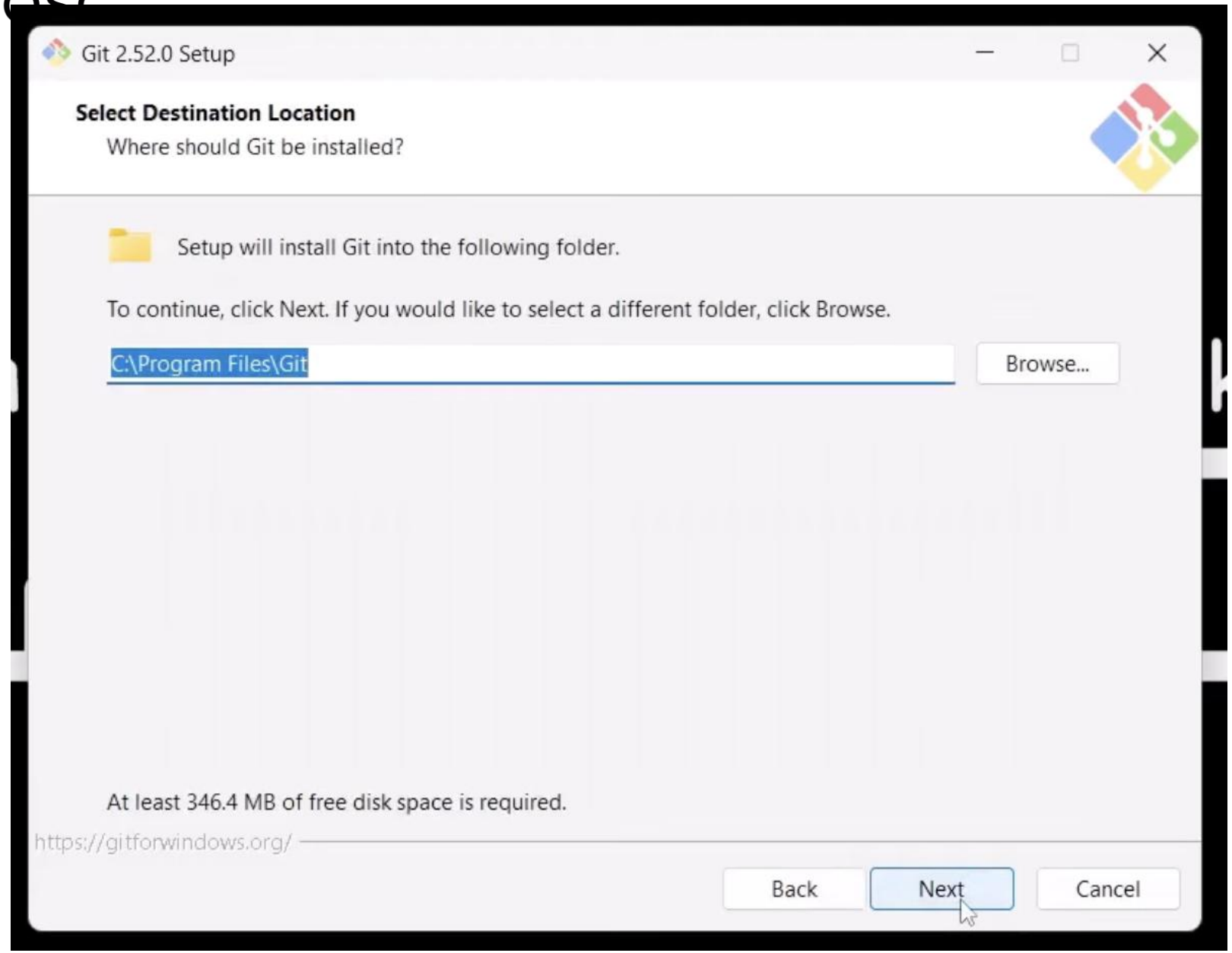
https://xkcd.com/1597/

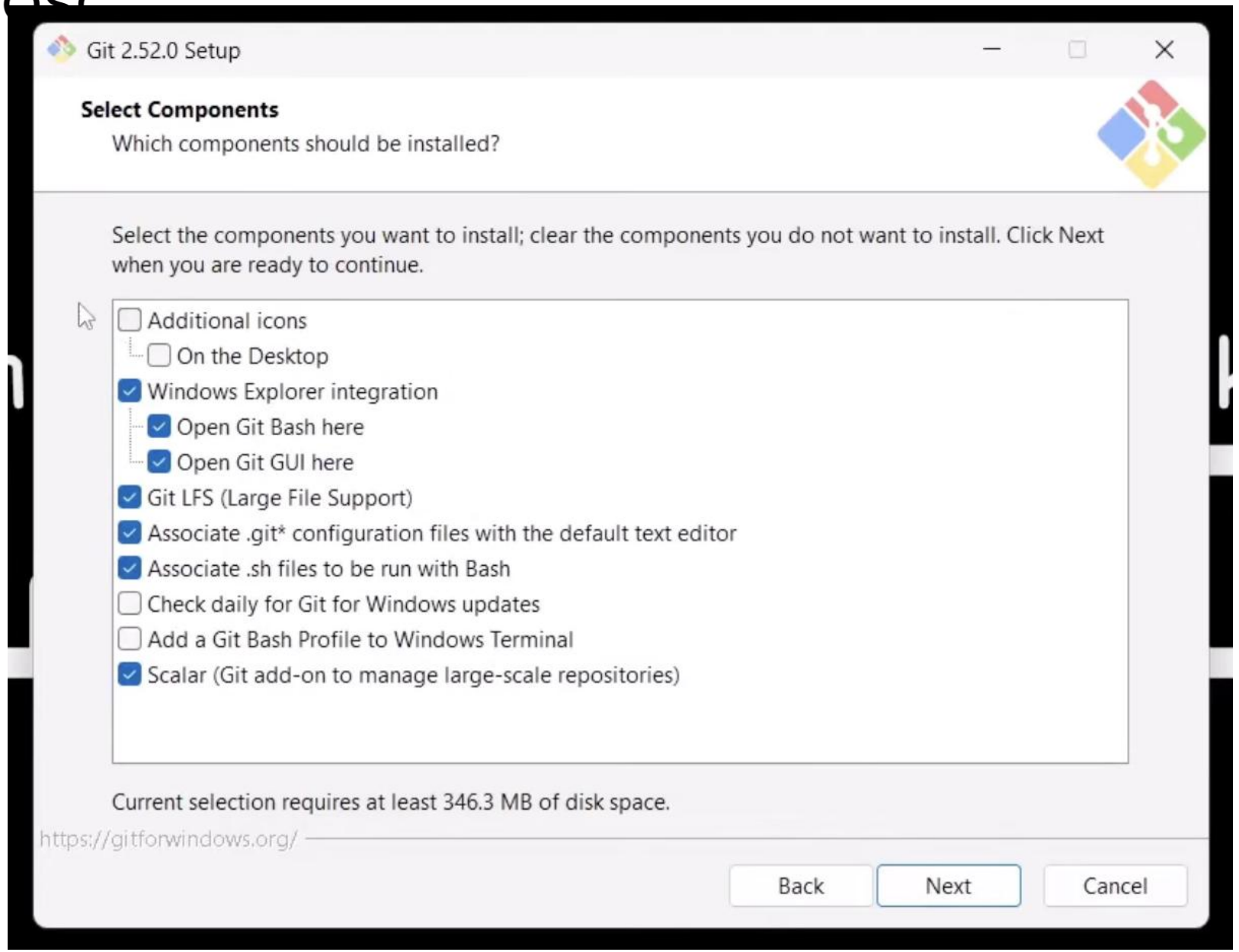
Instalace

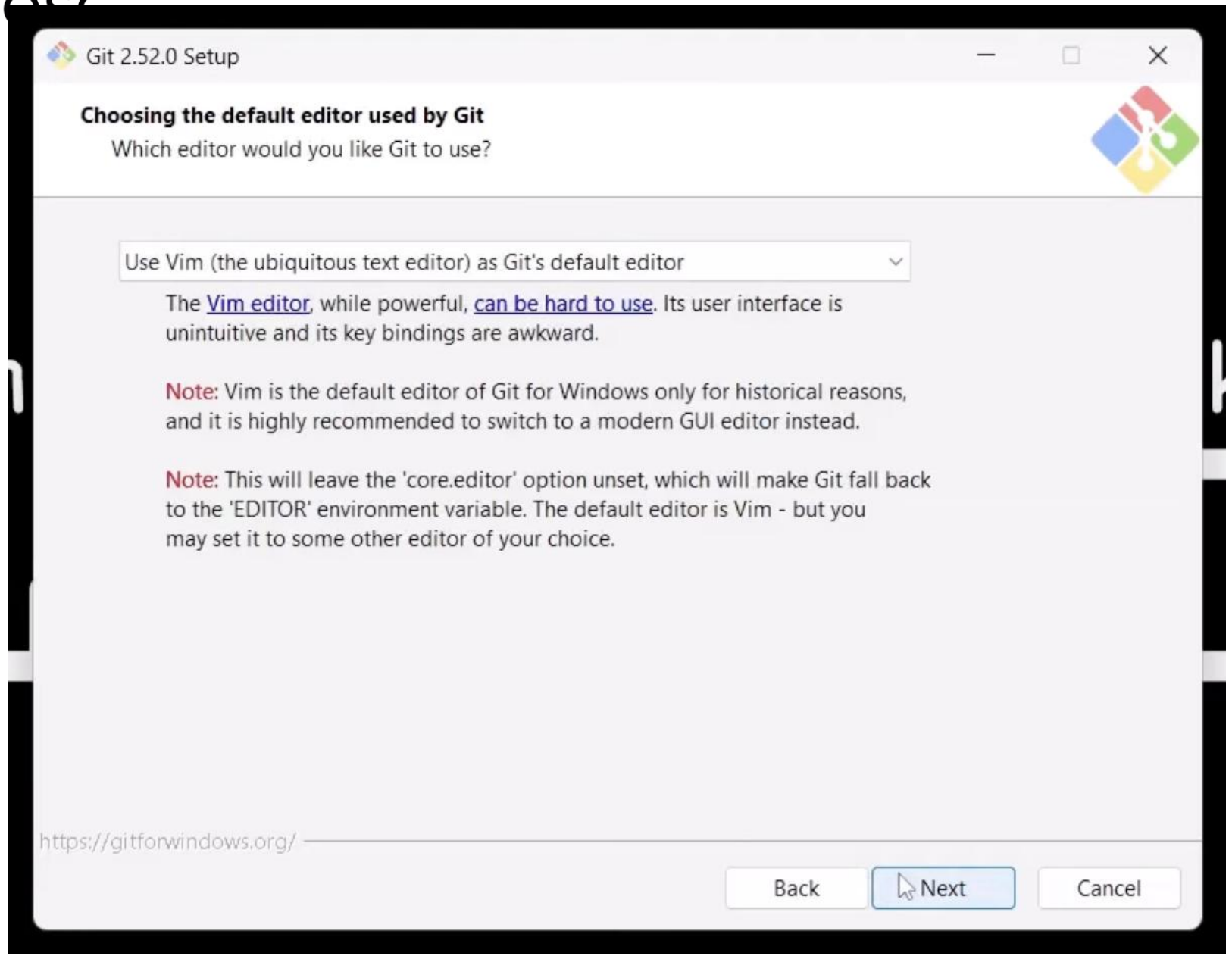
Windows

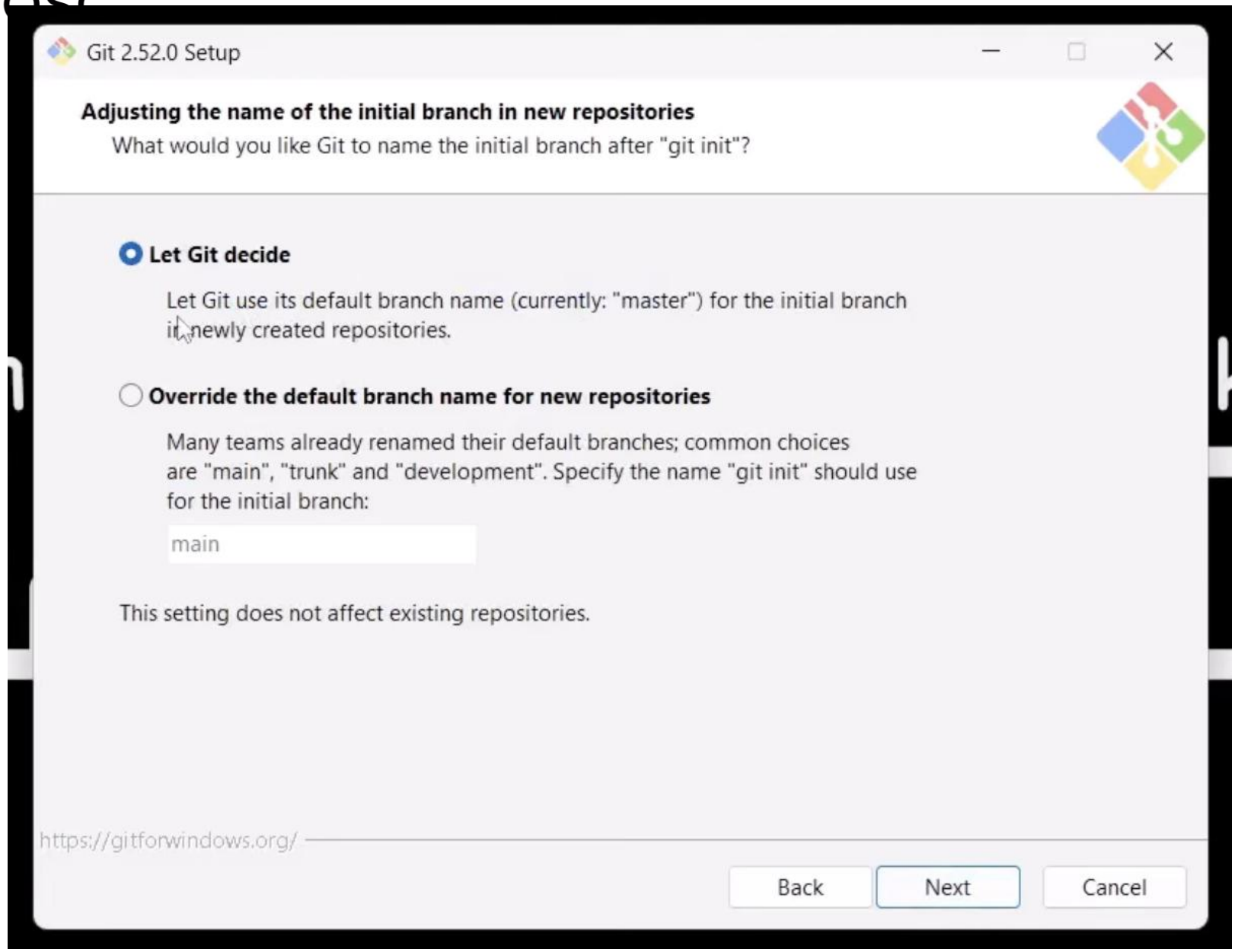
- git-scm.com/install/windows

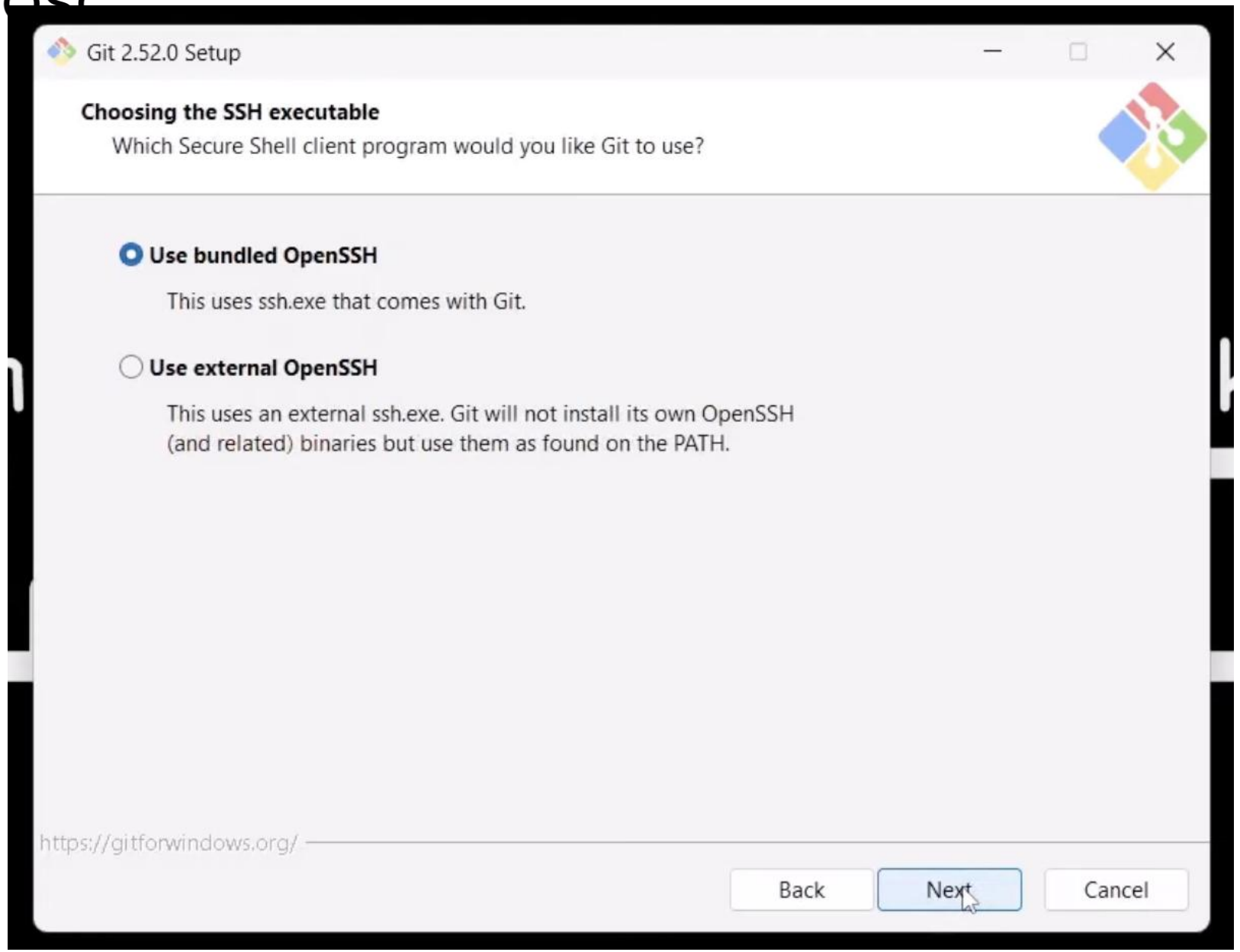


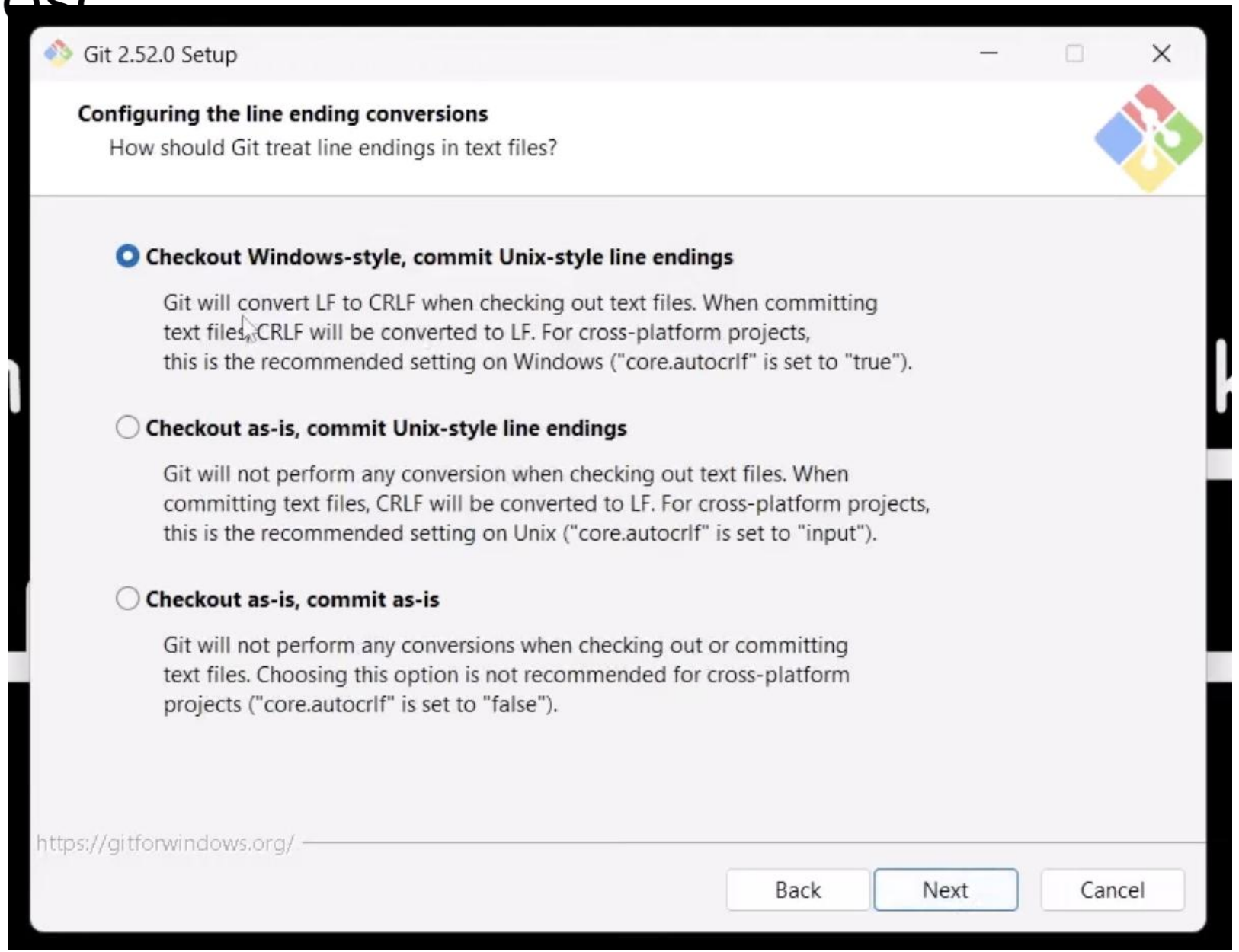

Git 2.52.0 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Git
Browse...
At least 346.4 MB of free disk space is required.
https://gitforwindows.org/
Back
Next
Cancel

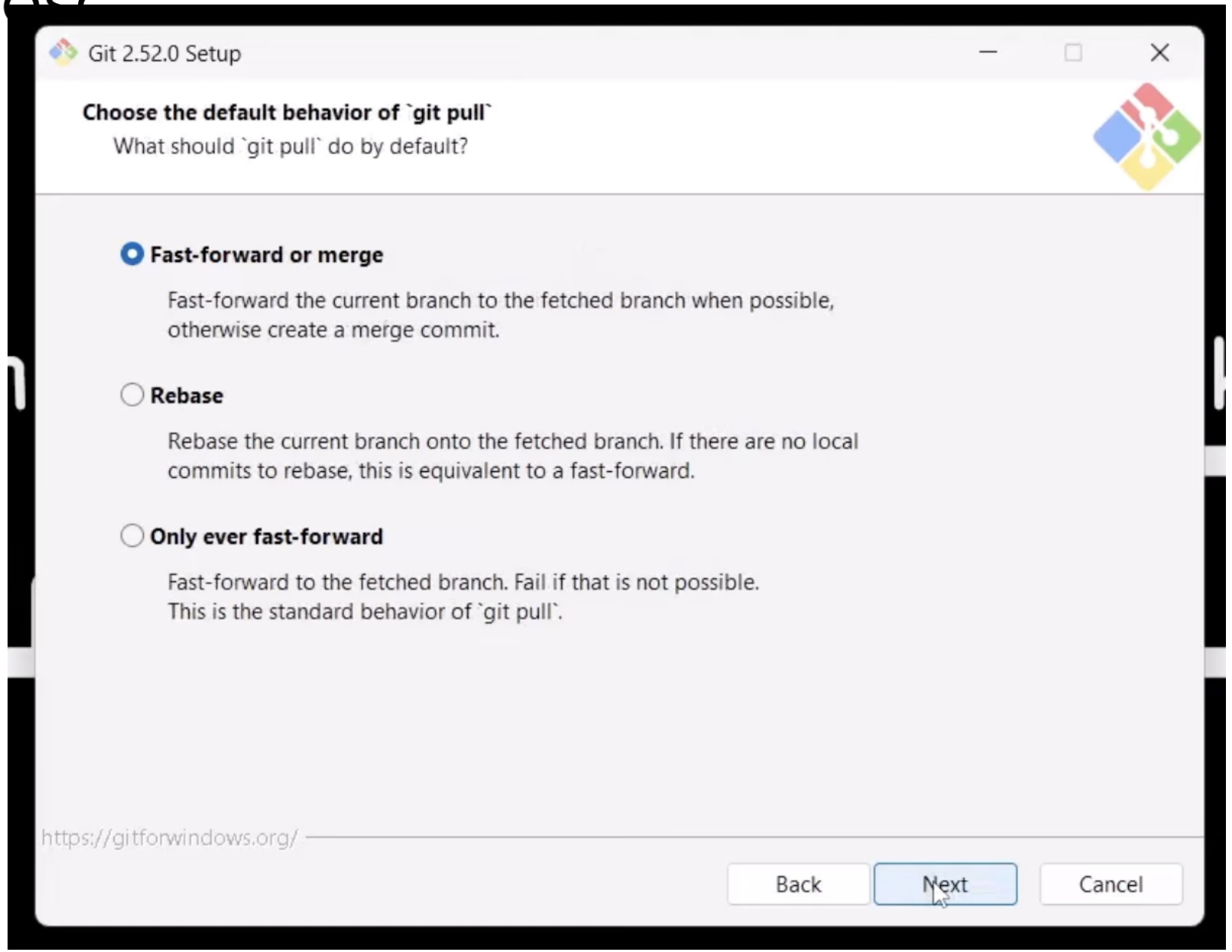

Git 2.52.0 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Open Git Bash here
Open Git GUI here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Check daily for Git for Windows updates
Add a Git Bash Profile to Windows Terminal
Scalar (Git add-on to manage large-scale repositories)
Current selection requires at least 346.3 MB of disk space.
https://gitforwindows.org/
Back
Next
Cancel

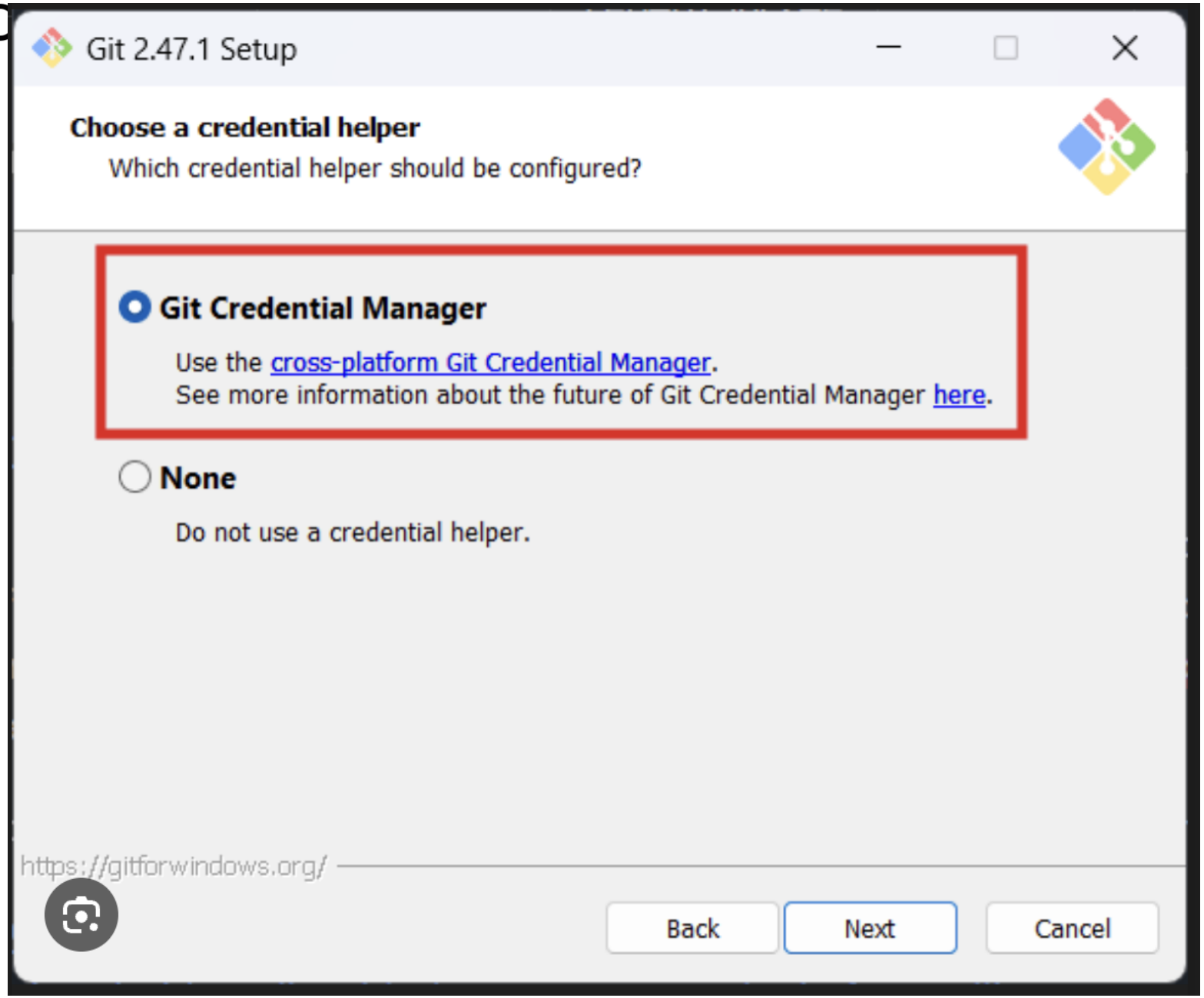

Git 2.52.0 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.
Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.52.0 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories.
Override the default branch name for new repositories
Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel

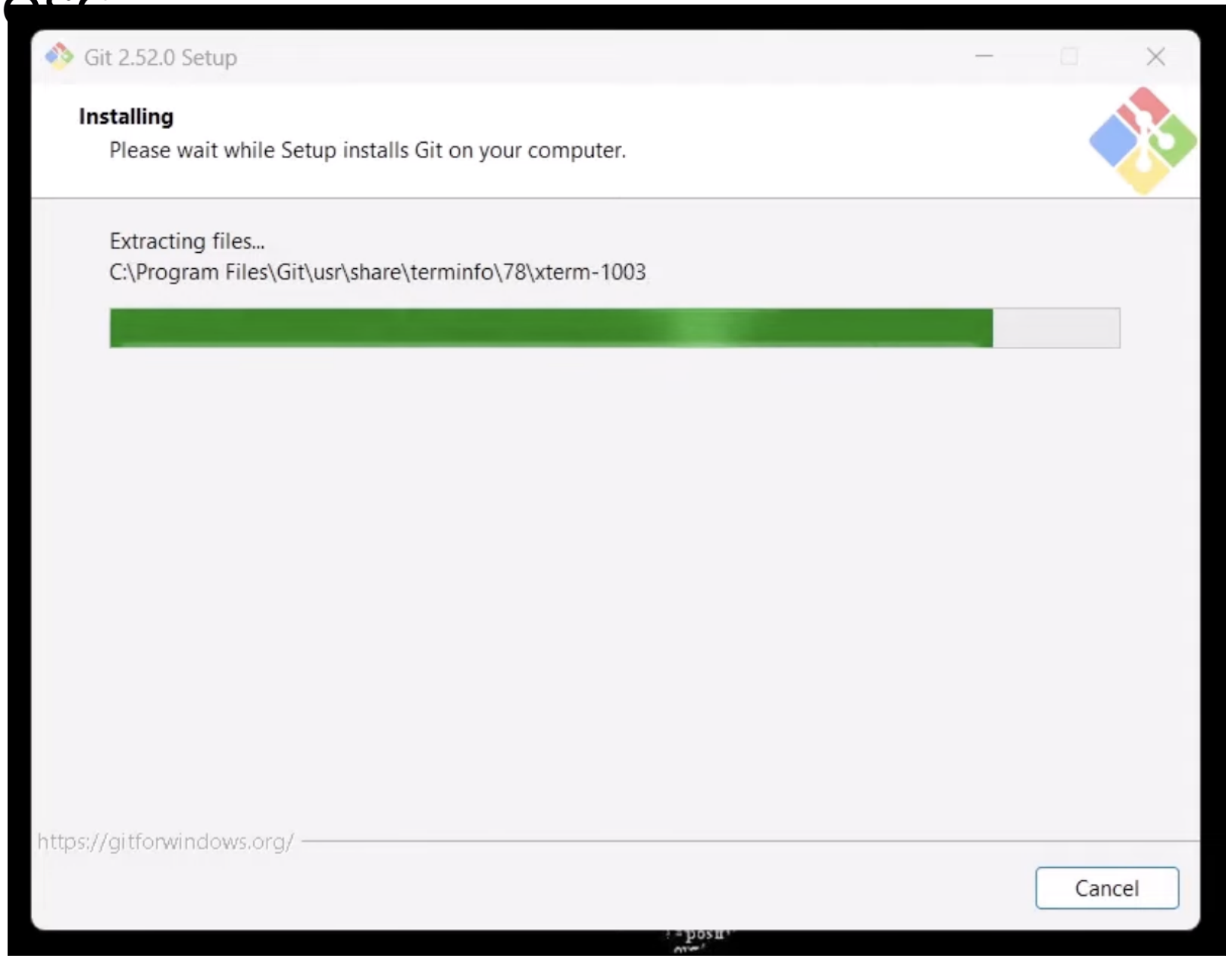

Git 2.52.0 Setup
Choosing the SSH executable
Which Secure Shell client program would you like Git to use?
Use bundled OpenSSH
This uses ssh.exe that comes with Git.
Use external OpenSSH
This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.
https://gitforwindows.org/
Back
Next
Cancel

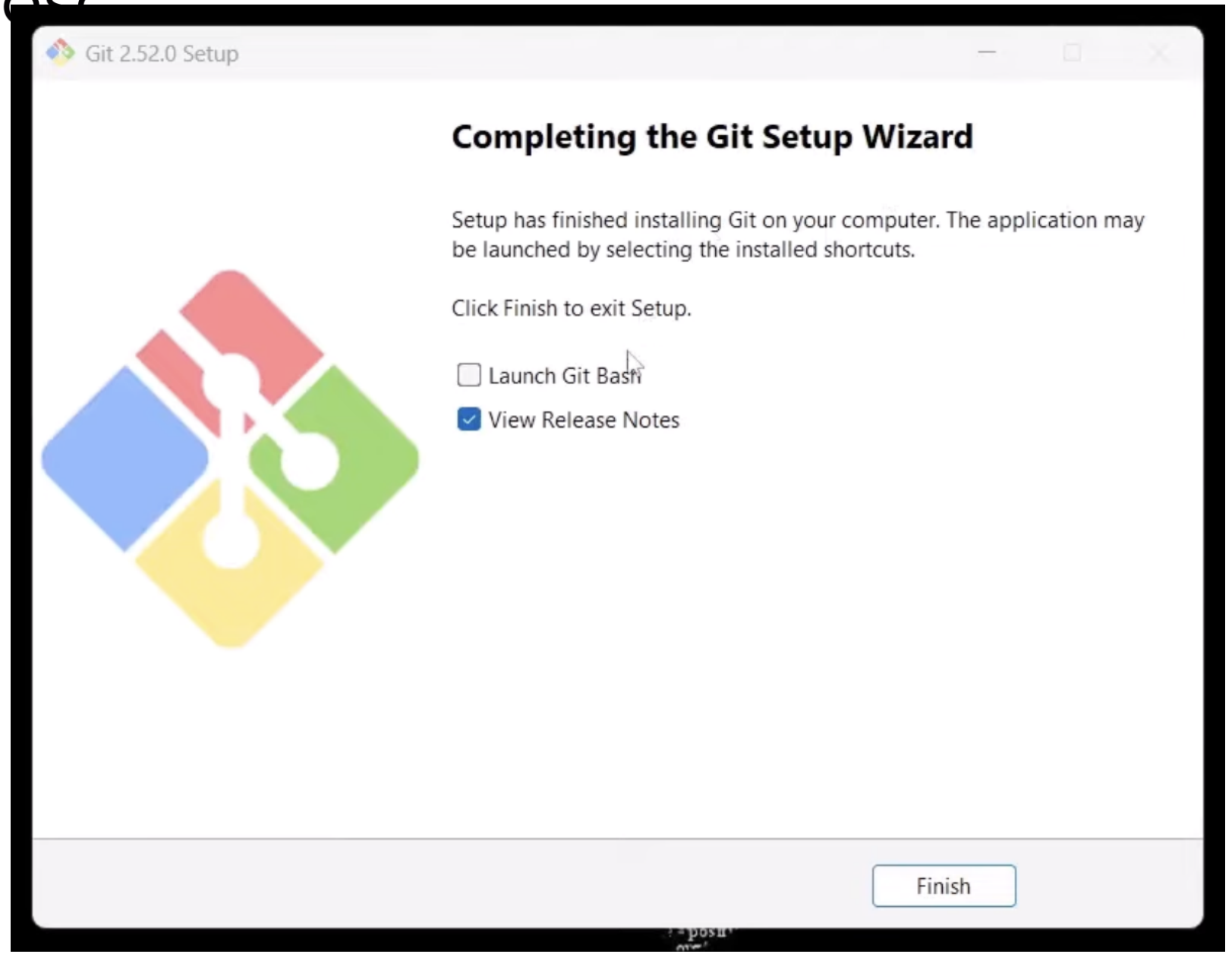

Git 2.52.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Back
Next
Cancel


Git 2.52.0 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Fast-forward or merge
Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.
https://gitforwindows.org/
Back
Next
Cancel

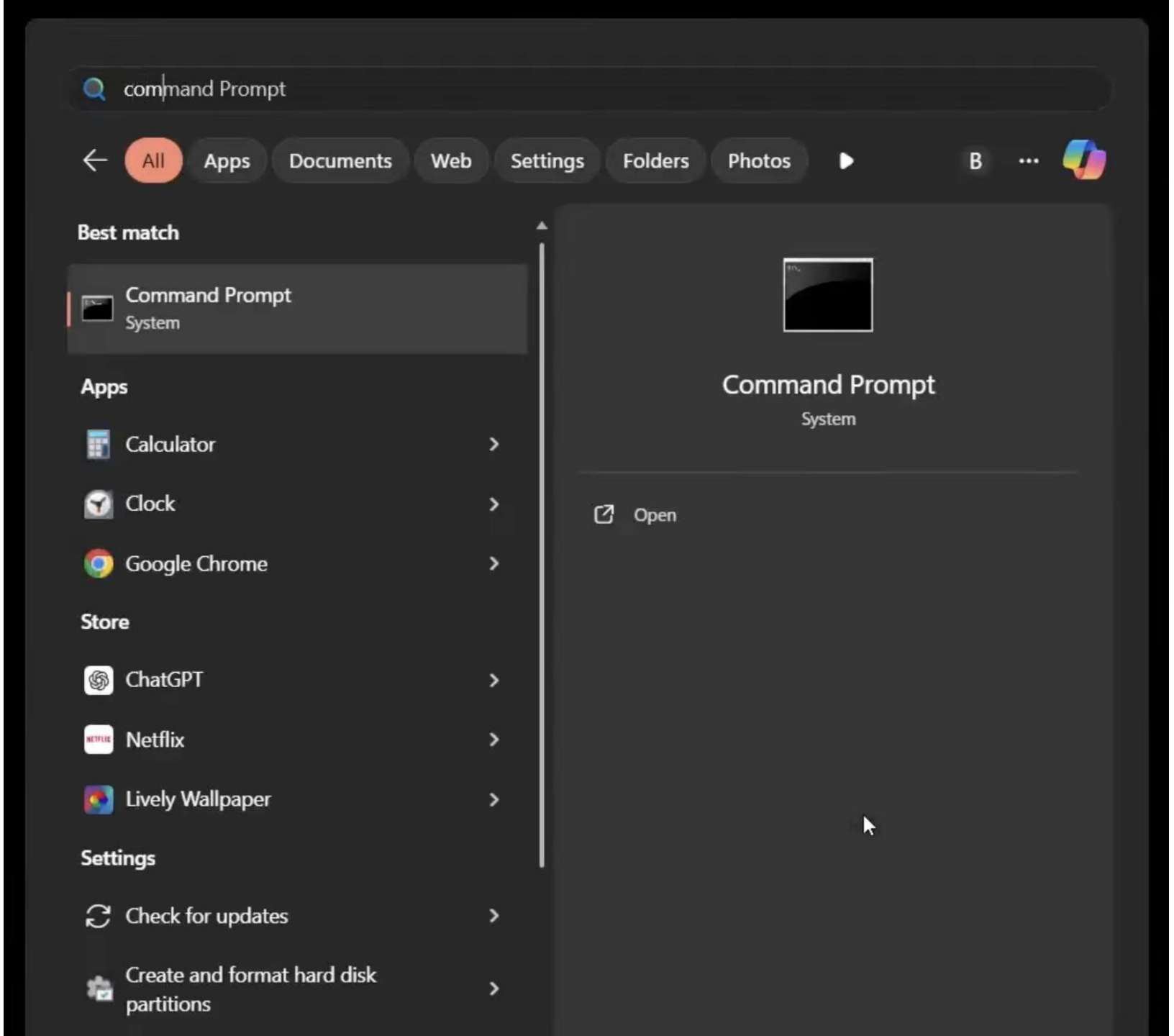

Git 2.47.1 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel

National Czech Programme




Git 2.52.0 Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Git Bash
View Release Notes
Finish

National Czech Programme



National Czech Programme | eosc

Instalace

Mac

- git-scm.com/install/mac

Install

Latest version: 2.53.0 (Release Notes)

Windows | macOS | Linux | Build from Source

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

Choose one of the following options for installing Git on macOS:

Homebrew

Install homebrew if you don't already have it, then:

```
$ brew install git
```

MacPorts

Install MacPorts if you don't already have it, then:

```
$ sudo port install git
```

Xcode Command Line Tools

Apple ships a binary package of Git with Xcode Command Line Tools. You can install this via:

```
$ xcode-select --install
```

Binary installer

Tim Harper provided an installer for Git until version 2.33.0 / 2021. These installers are no longer linked from here because there are no updates since that version, nor are there plans to provide any.

Installing git-gui

If you would like to install git-gui and gitk, git's commit GUI and interactive history browser, you can do so using homebrew

```
$ brew install git-gui
```

National Czech Programme | eosc

Instalace

Mac

- git-scm.com/install/mac

nejjedndussi cesta instalace git na macos

Nejjednodušší cesta na macOS je fakt na 2 kliky — Apple to má dobře vyřešené 👇

✅ Varianta 1 (doporučená): přes Xcode Command Line Tools

Stačí do terminálu napsat:

```bash
git --version
```

👉 Pokud Git nemáš, macOS ti **sám nabídne instalaci**
→ klikneš **Install**
→ hotovo za pár minut

Tohle je nejrychlejší a čisté řešení.

✅ Varianta 2: Homebrew (pokud ho používáš)

Pokud máš Homebrew:

```bash
brew install git
```

👉 Výhoda:

- máš vždy aktuální verzi
- lepší pro dev workflow

Po instalaci

```
git --version
git --help
```

Konfigurace

```
git --list
git config --global user.name
git config --global user.email
```

Konfigurace

Jak najít umístění configu?

- `git config --list --show-origin`

Lokální → platí jen pro jeden konkrétní projekt

`{projekt}\.git\config`

Globální → platí pro všechny repozitáře daného uživatele

`C:\Users\{user}\.gitconfig`

Systémový → platí pro celý počítač (všichni uživatelé)

`C:\Program Files\Git\etc\gitconfig`

Inicializace projektu

Vytvoření projektu

- `mkdir nazev-projektu`
- `cd projekt`
- **`git init`**

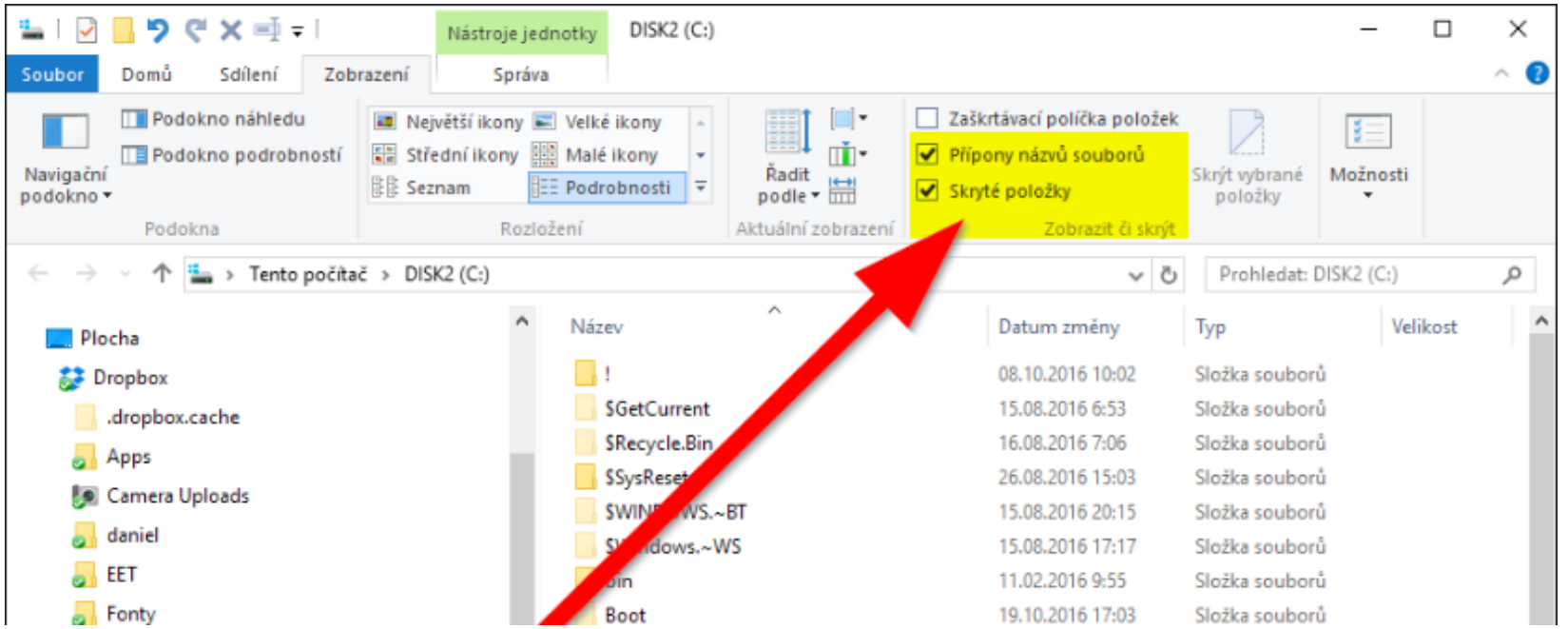
Inicializace projektu

Vytvoření projektu

- `mkdir nazev-projektu`
- `cd projekt`
- **`git init`**

```
marek@Mac EOSC % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/marek/Desktop/EOSC/.git/
```

Inicializace projektu



https://365tipu.cz/2016/11/13/jak-ve-windows-zobrazit-skryte-soubory/

Inicializace projektu

Změna názvu hlavní větvě

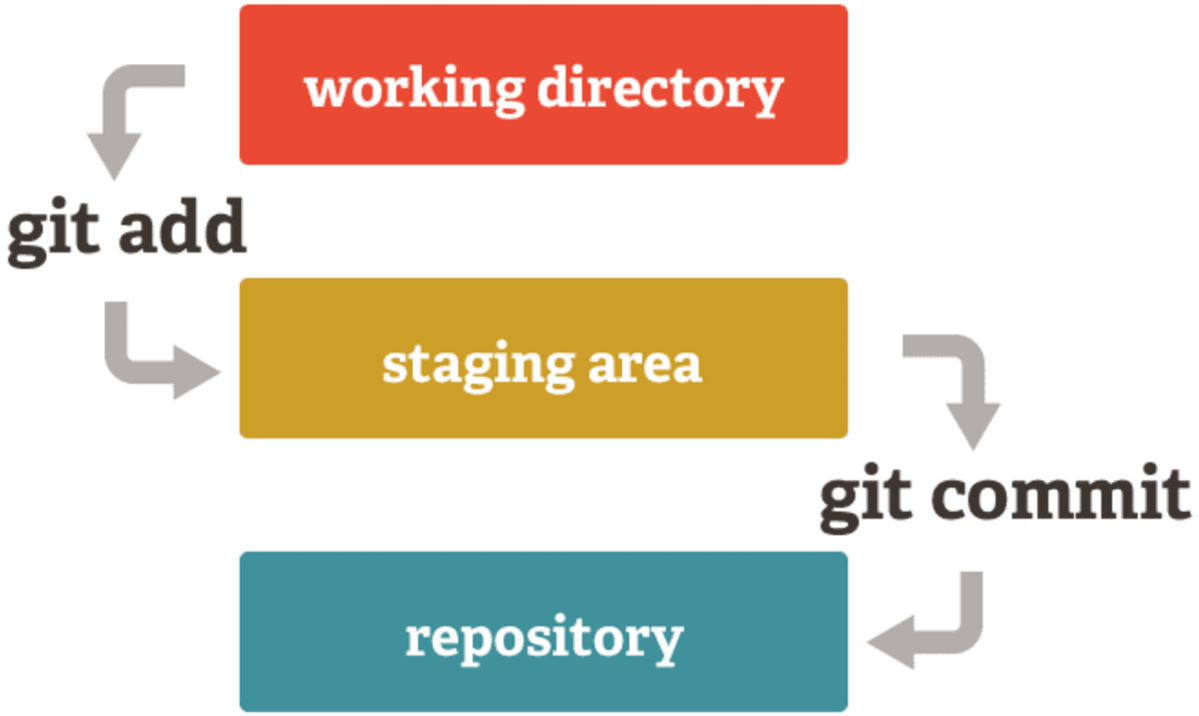
- `git branch -m main` → **lokální**
- `git config --global init.defaultBranch main` → **globální**

Aktuální stav projektu

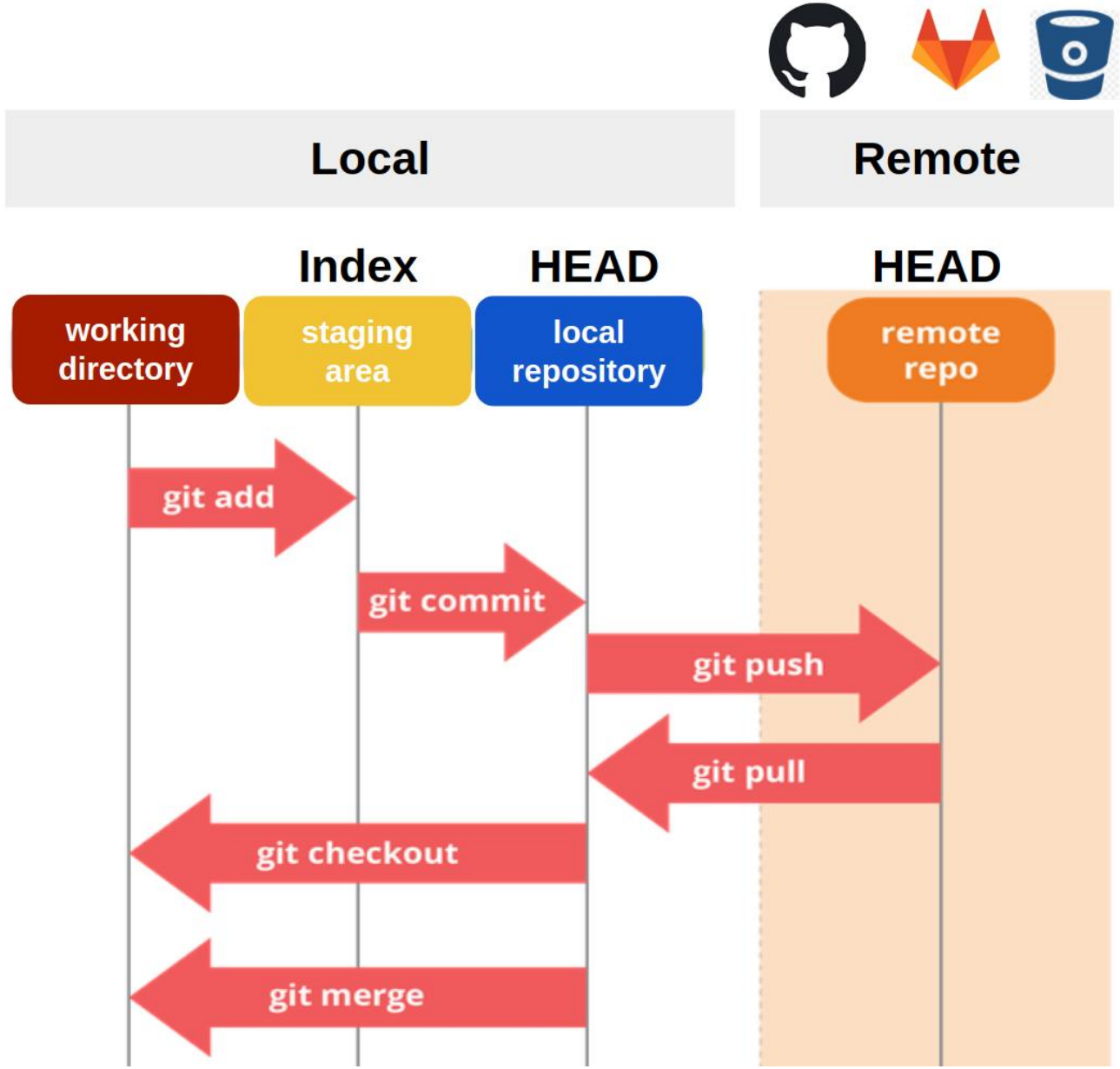
- `git status`

```
marek@Mac EOSC % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   soubor-staged.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        soubor-untracked.txt
```

Kroky / Stavy



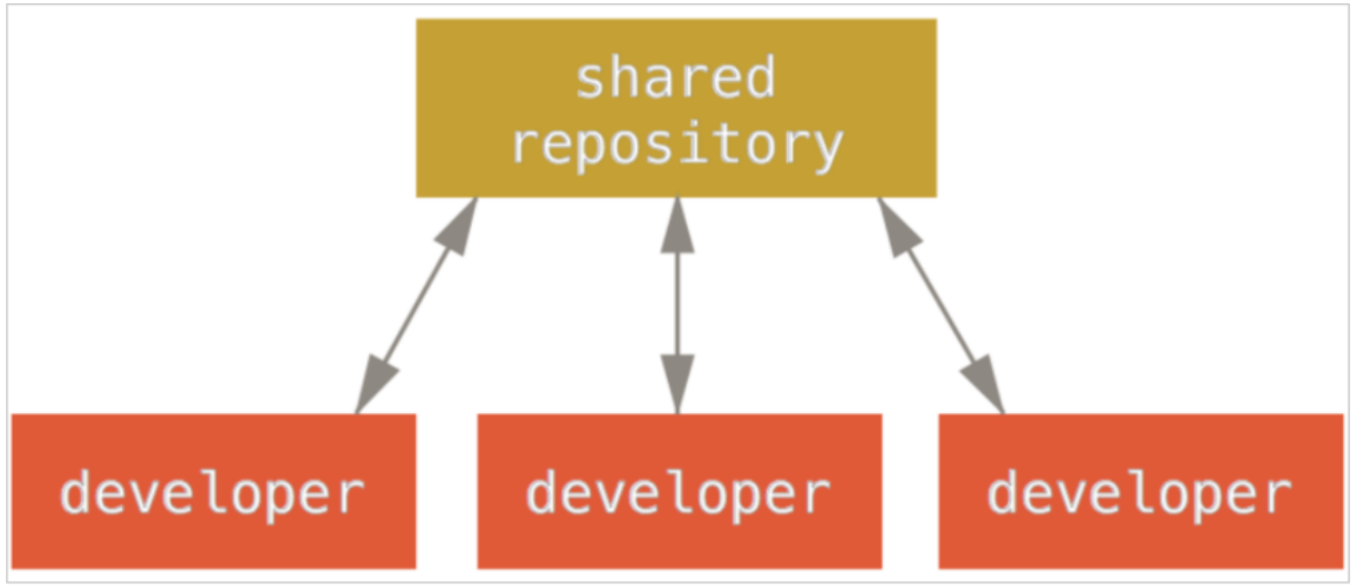
https://stopbyte.com/t/what-is-git-how-to-use-git/994



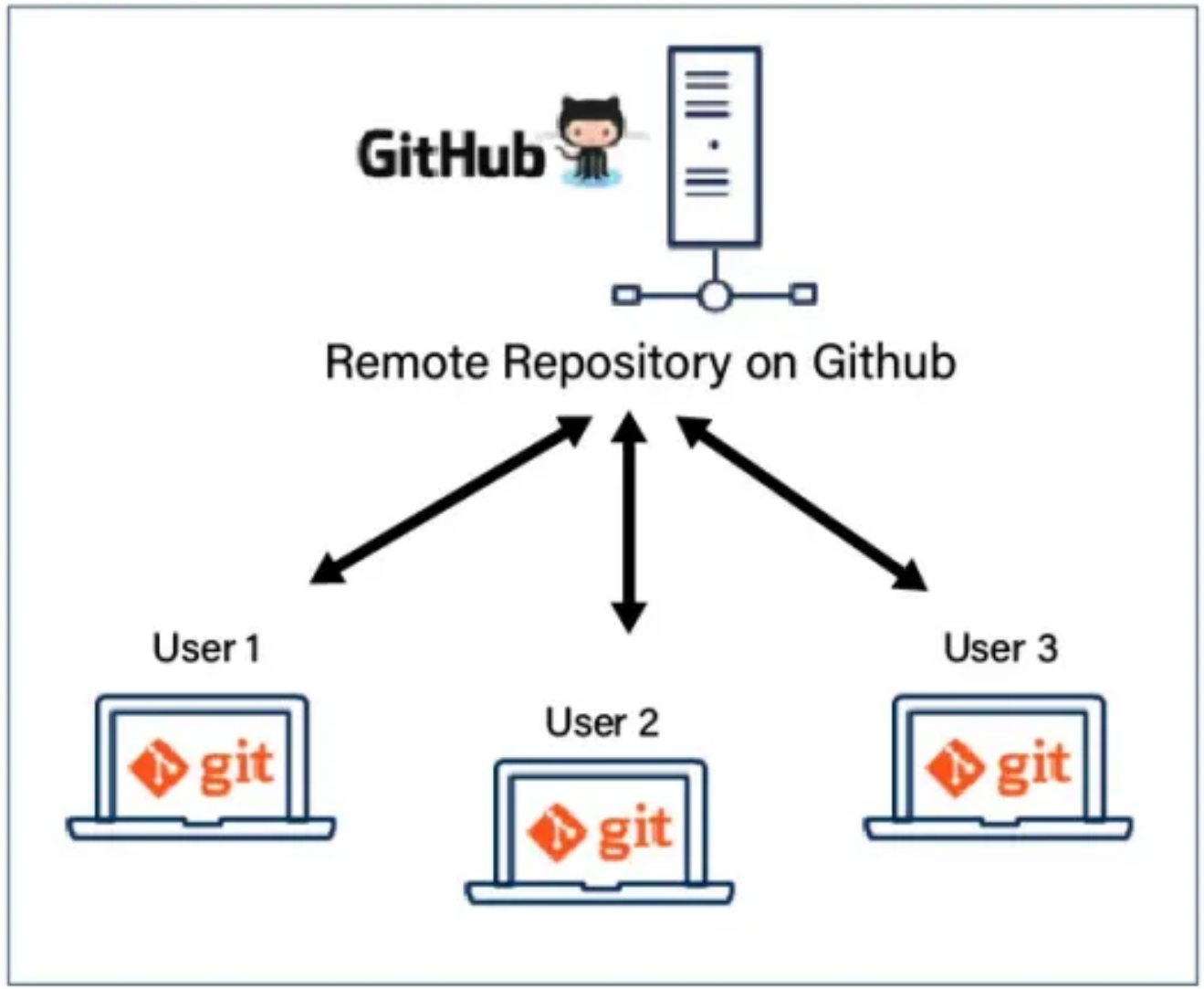
https://huogerac.hashnode.dev/most-useful-git-commands-by-git-stages

Workflow

Centralizovaný pracovní postup

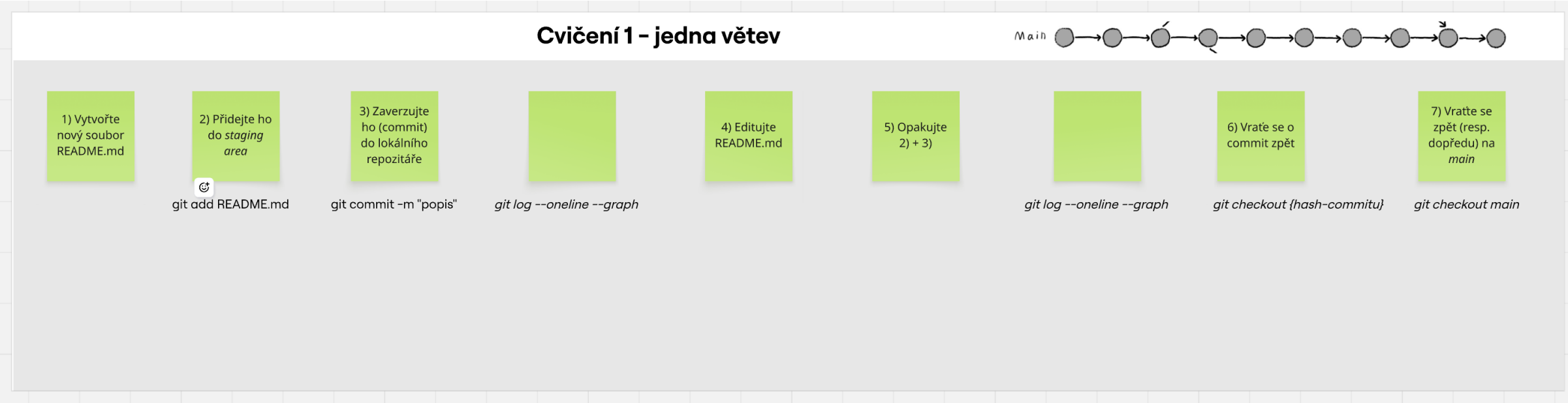


https://git-scm.com/book/id/v2/Distributed-Git-Distributed-Workflows



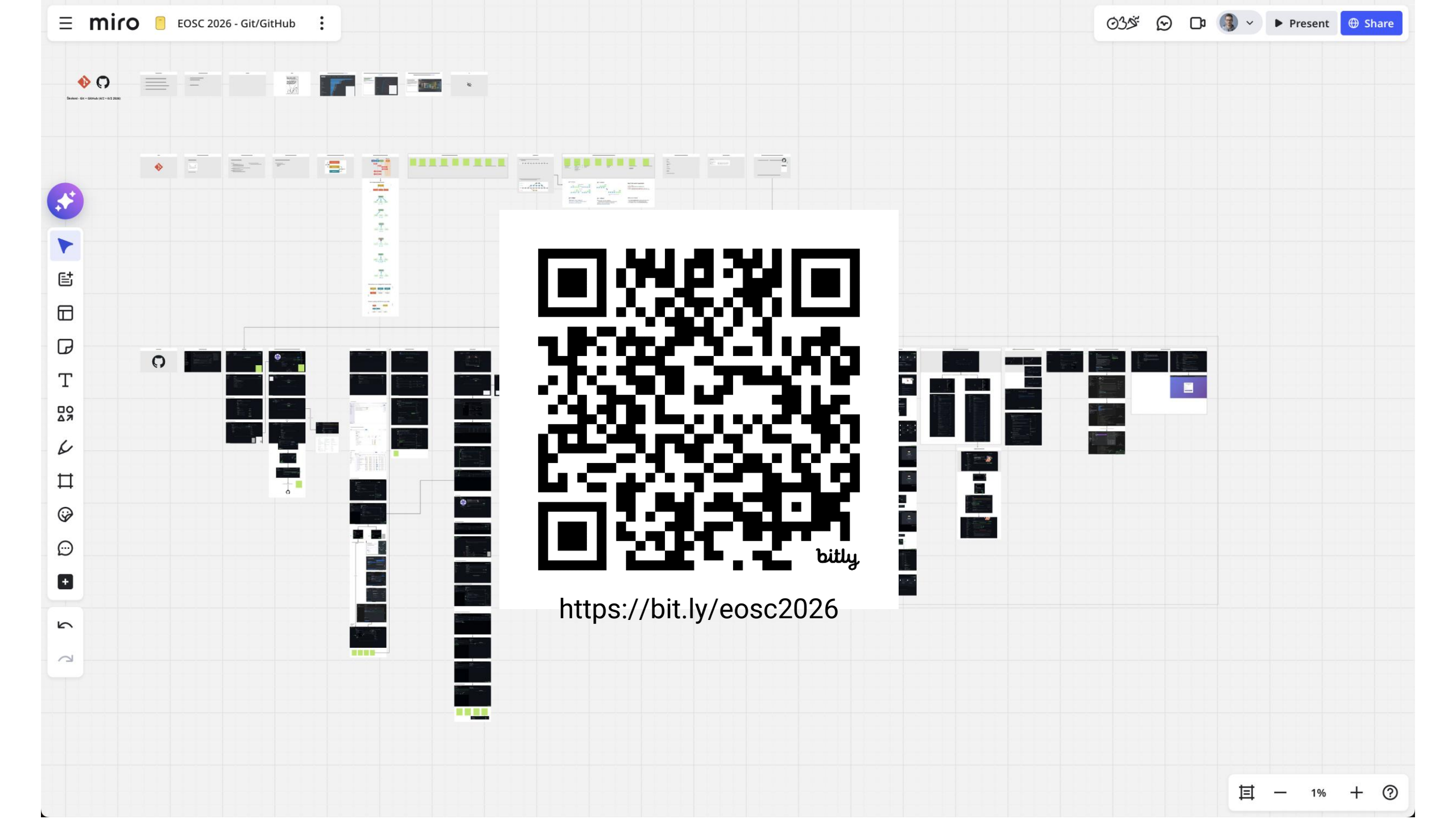
https://www.sprintzeal.com/blog/how-to-use-github

Cvičení 1


Cvičení 1 - jedna větev
Main
1) Vytvořte nový soubor README.md
2) Přidejte ho do staging area
git add README.md
3) Zaverzujte ho (commit) do lokálního repozitáře
git commit -m "popis"
git log --oneline --graph
4) Editujte README.md
5) Opakujte 2) + 3)
git log --oneline --graph
6) Vraťe se o commit zpět
git checkout {hash-commitu}
7) Vraťte se zpět (resp. dopředu) na main
git checkout main

1. Cvičení

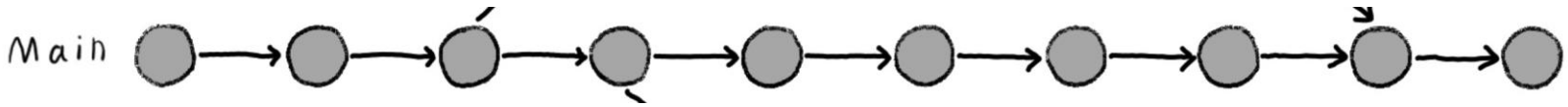
https://bit.ly/eosc2026


miro
EOSC 2026 - Git/GitHub
Present
Share
bitly
https://bit.ly/eosc2026
1%

Větvení

Přepnutí na existující větev / commit (stav) → **již známe**

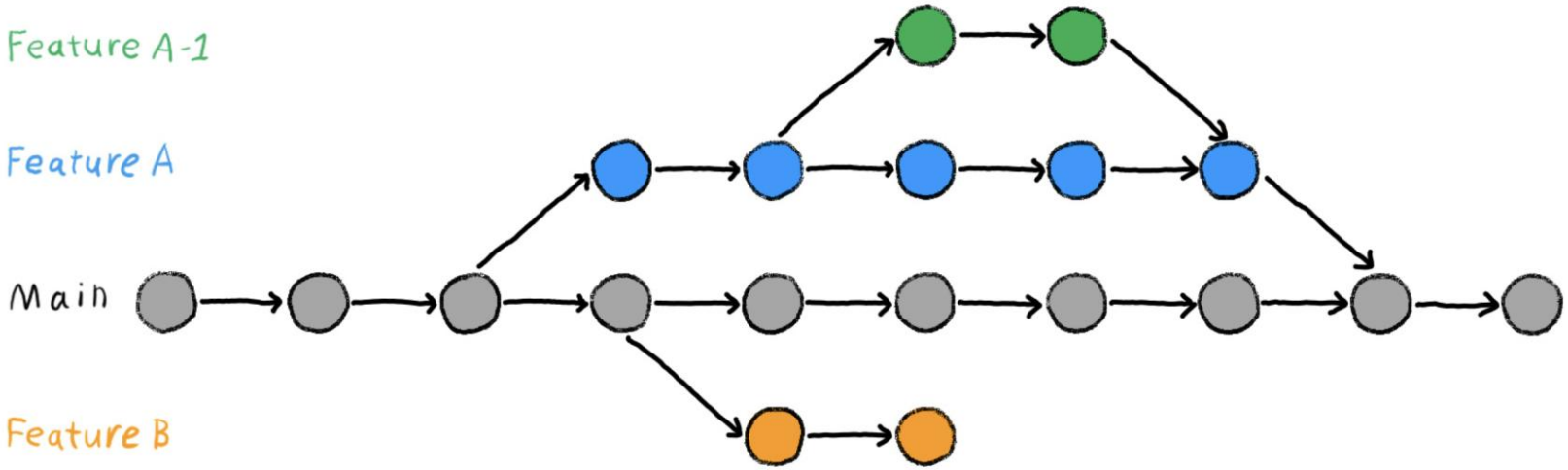
- `git checkout`



Větvení

Vytvoření nové větvě a přepnutí do jejího kontextu (na ní)

- `git checkout -b {nazev-nove-lokalni-vetve}`



Sloučení

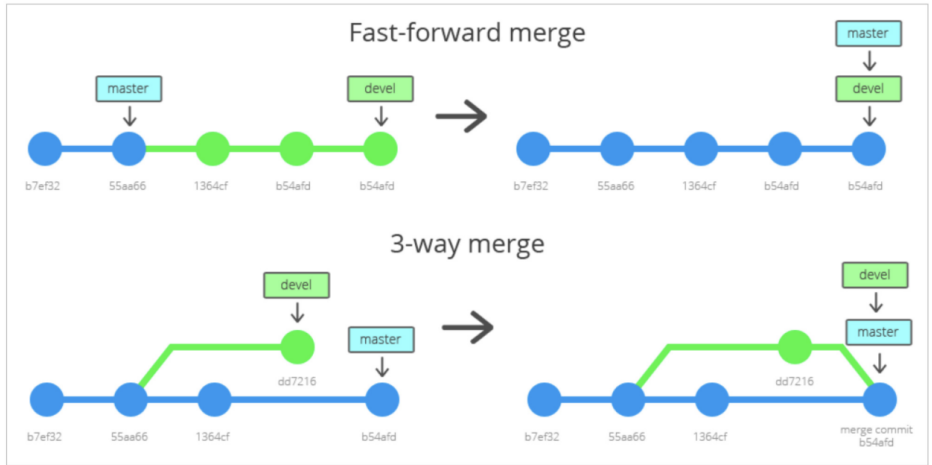
Sloučení dvou větví do jedné

- `git merge {nazev-vetve-kterou-chci-sloucit}`

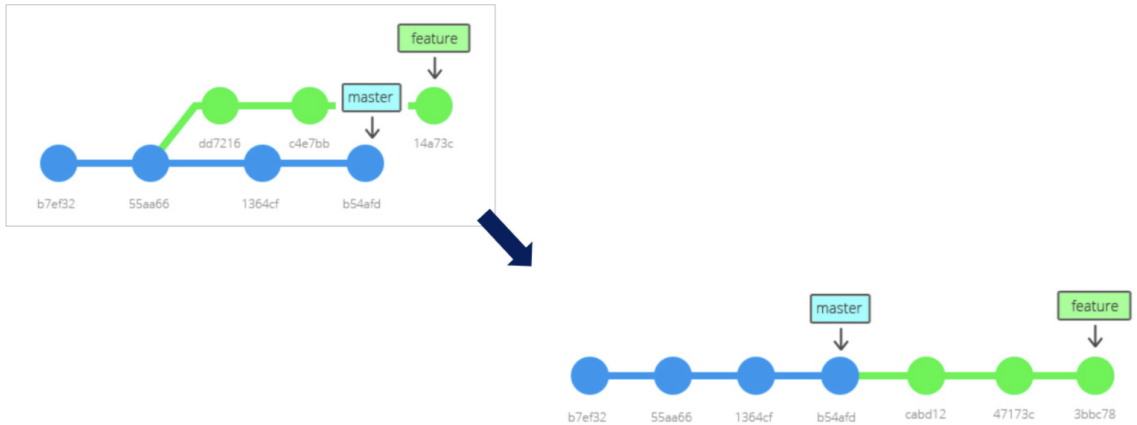
(nebo)

- `git rebase {nazev-vetve-na-kterou-chci-prestavet}`

git merge



git rebase



commity mají jiný hash, po dokončení staré commity smaže

git merge

- Sloužení větví - jako nový nový *commit*
- *"Joins two or more development histories together"*
- https://git-scm.com/docs/git-merge

git rebase

- Sloučení větví - změna báze (základny)
  - *Commity* se přesunou, změní pořadí, stlačí se do jediného apod.
- *"Reapplies commits on top of another base branch"*
- https://git-scm.com/docs/git-rebase

Konflikty

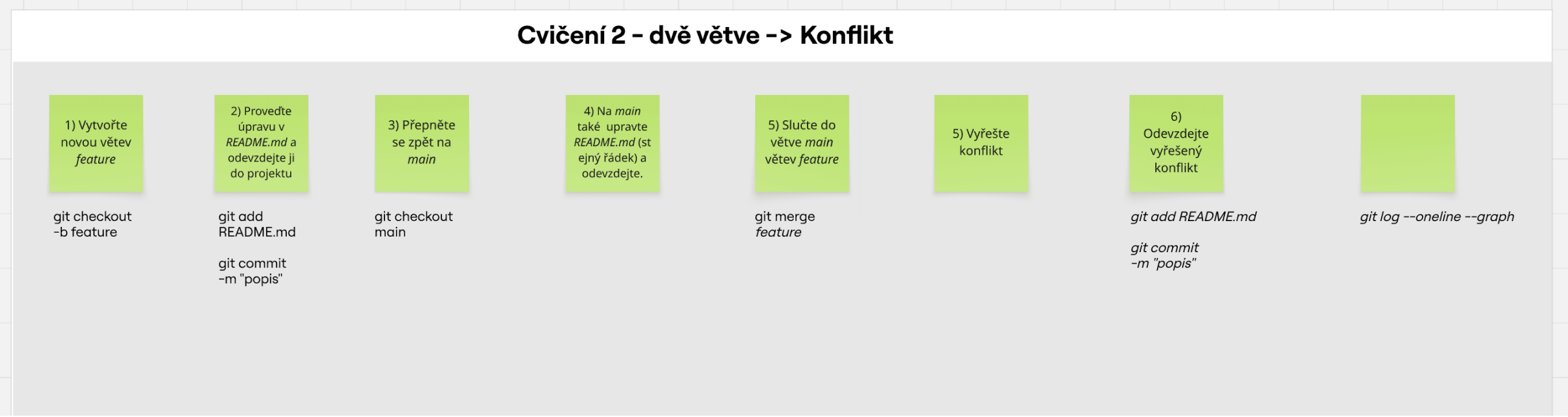
Když 2 lidé změní stejný řádek

```
<<<<<<< HEAD
function myAwesomeFunction($params){
=======
function myAwesomeFunction($format, $params){
>>>>>>> e28fbd0a3b987a0d700a79d1d19ae73bf39eff2d
```

National Czech Programme | eosc

git

2. Cvičení



https://bit.ly/eosc2026

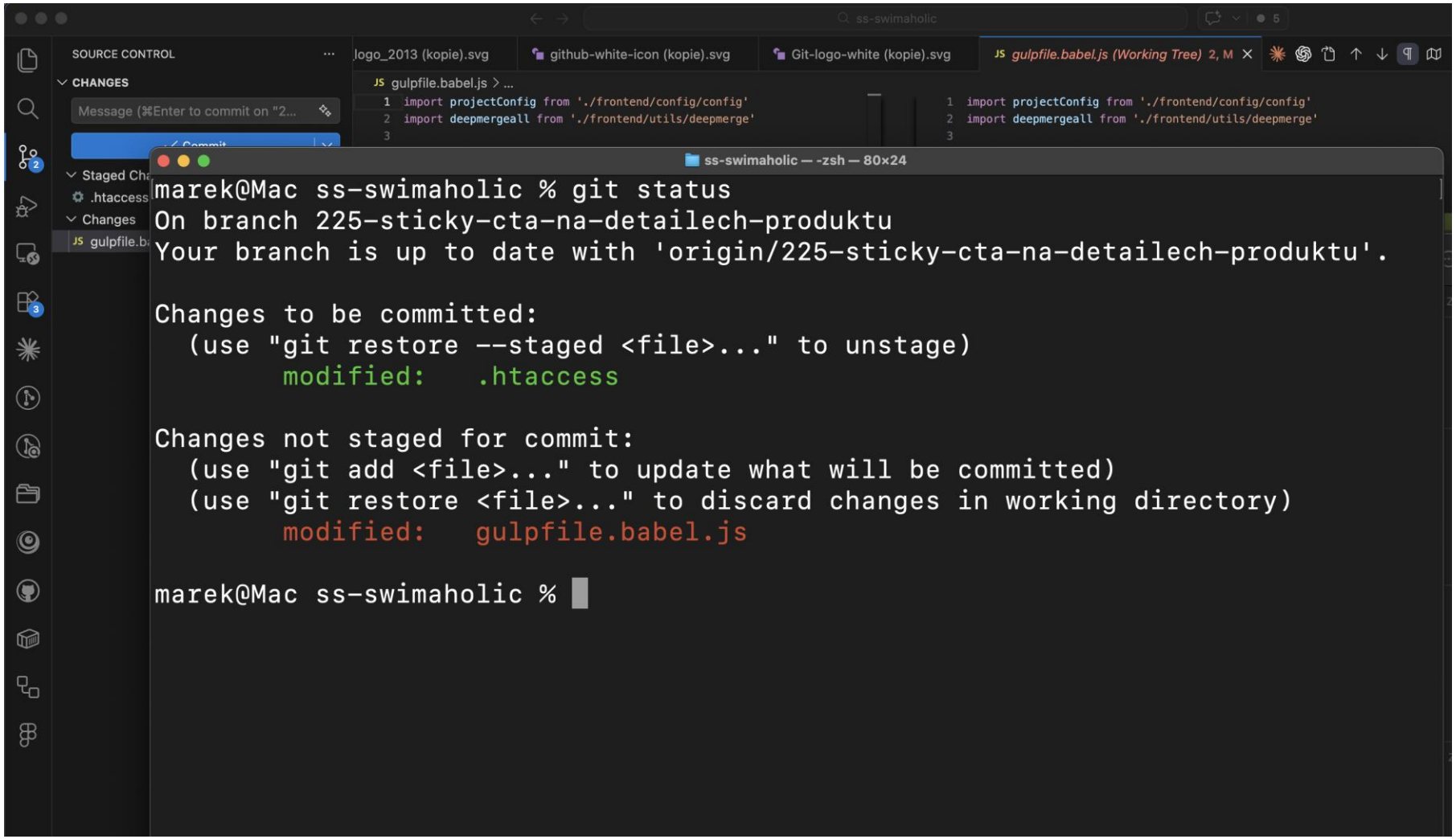
Nejčastější příkazy

- `status`
- `add`
- `commit`
- `checkout`
- `merge`

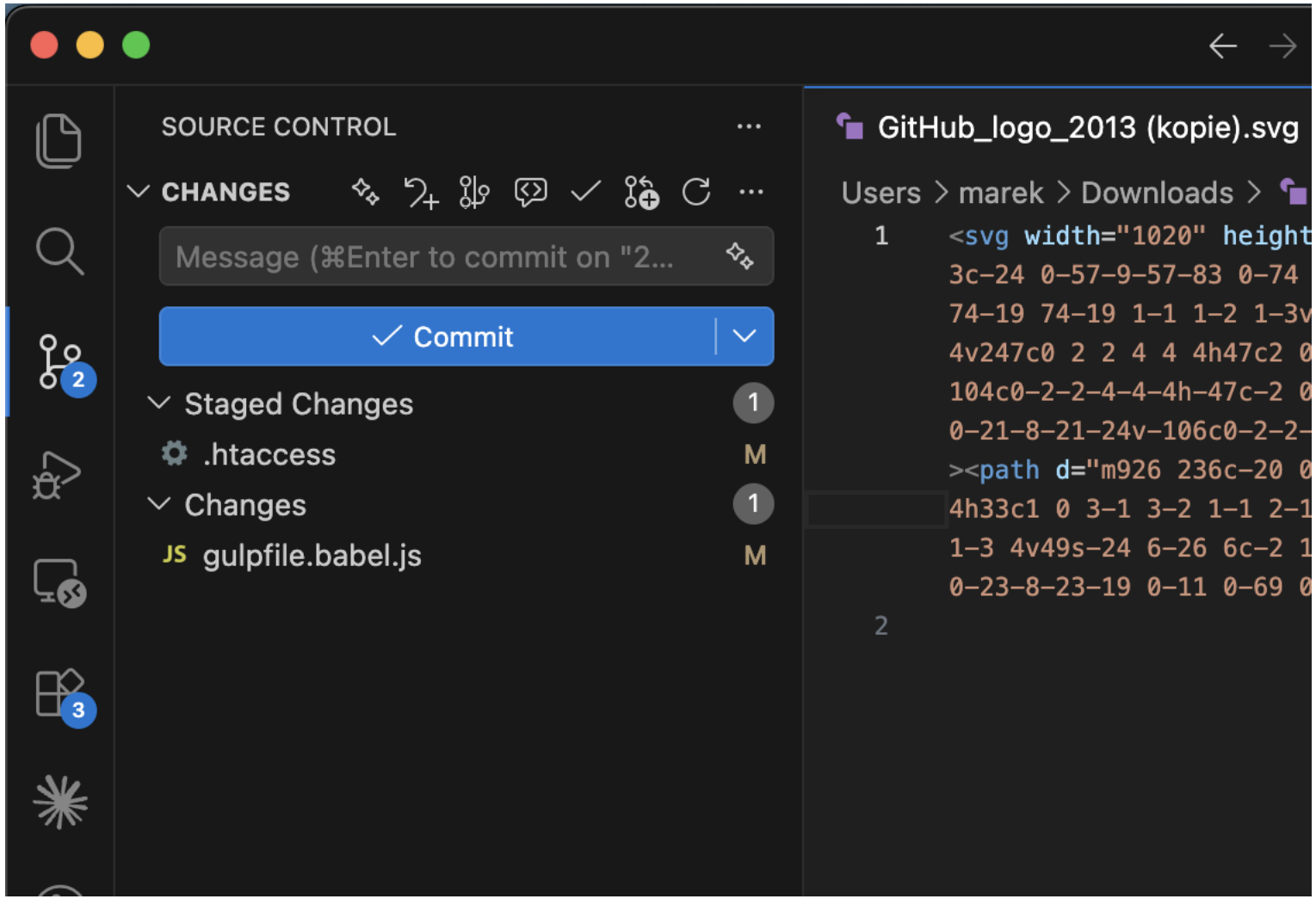
- `pull (fetch)`
- `push`

- `git log`
- `git diff`
- `git stash (list / apply)`

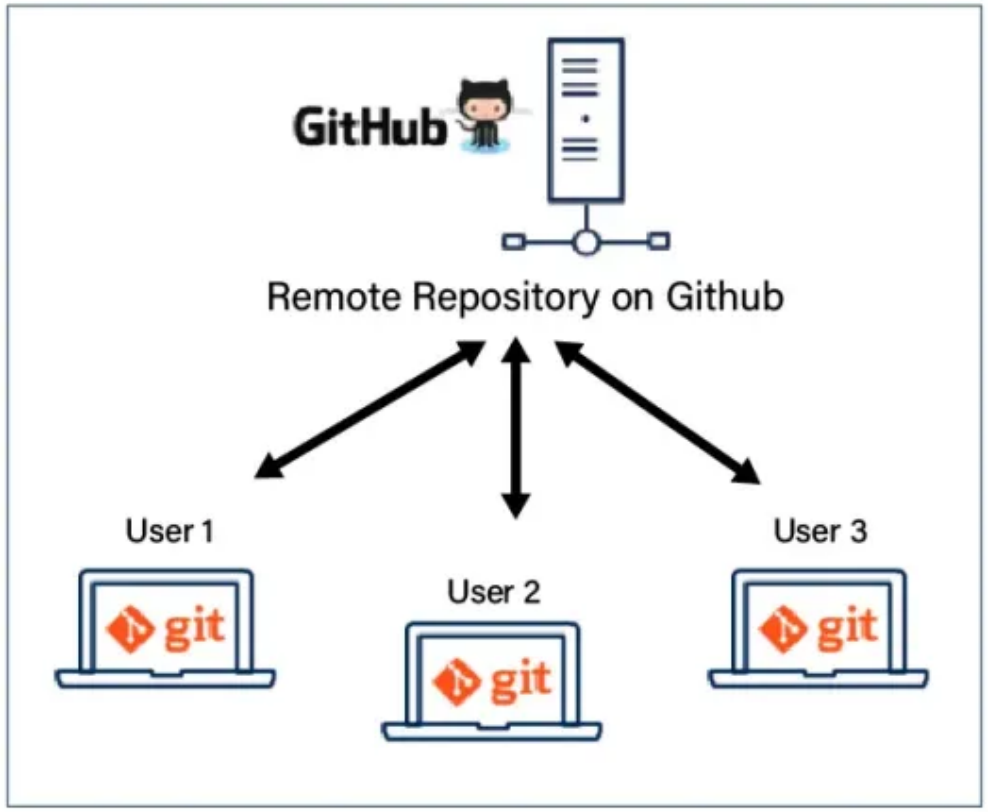
CLI vs GUI


SOURCE CONTROL
CHANGES
Message (⌘Enter to commit on "2...
Staged Changes
.htaccess
Changes
gulpfile.babel.js
logo_2013 (kopie).svg
github-white-icon (kopie).svg
Git-logo-white (kopie).svg
gulpfile.babel.js (Working Tree) 2, M
import projectConfig from './frontend/config/config'
import deepmergeall from './frontend/utils/deepmergeall'
ss-swimaholic — -zsh — 80×24
marek@Mac ss-swimaholic % git status
On branch 225-sticky-cta-na-detailech-produktu
Your branch is up to date with 'origin/225-sticky-cta-na-detailech-produktu'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: .htaccess
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: gulpfile.babel.js
marek@Mac ss-swimaholic %

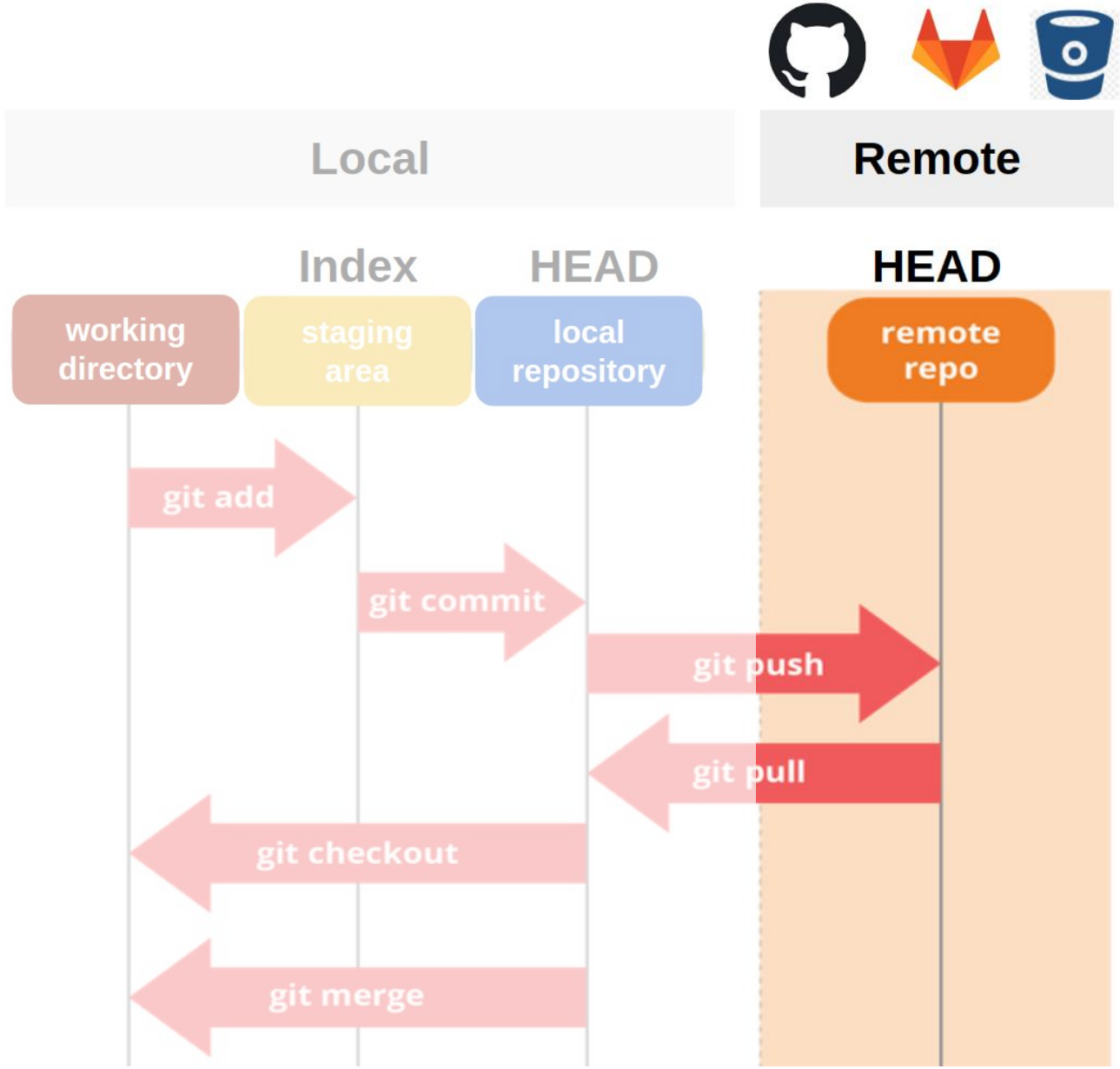
CLI vs GUI


SOURCE CONTROL
CHANGES
Message (⌘Enter to commit on "2...
Commit
Staged Changes 1
.htaccess M
Changes 1
gulpfile.babel.js M
GitHub_logo_2013 (kopie).svg
Users > marek > Downloads >
1 <svg width="1020" height
3c-24 0-57-9-57-83 0-74
74-19 74-19 1-1 1-2 1-3v
4v247c0 2 2 4 4 4h47c2 0
104c0-2-2-4-4-4h-47c-2 0
0-21-8-21-24v-106c0-2-2-
><path d="m926 236c-20 0
4h33c1 0 3-1 3-2 1-1 2-1
1-3 4v49s-24 6-26 6c-2 1
0-23-8-23-19 0-11 0-69 0
2

Remote origin



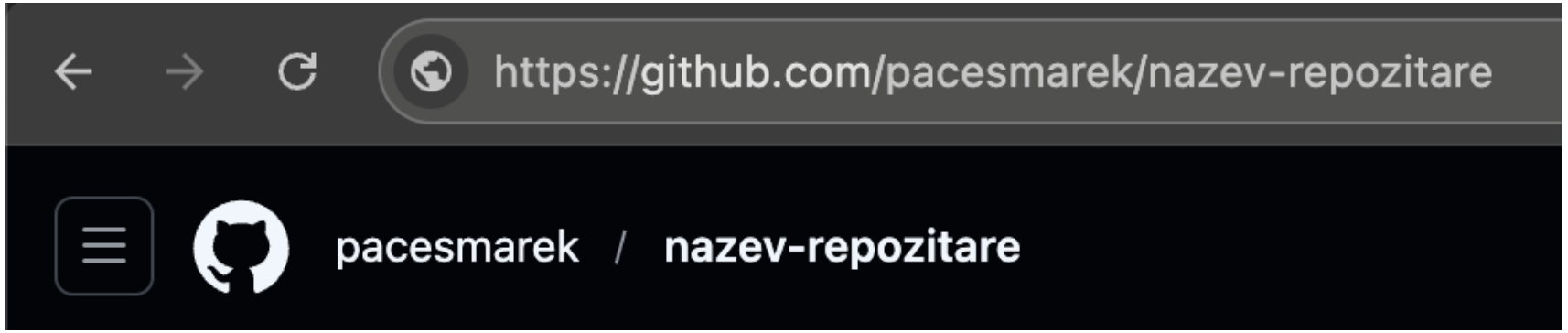
https://www.sprintzeal.com/blog/how-to-use-github



https://huogerac.hashnode.dev/most-useful-git-commands-by-git-stages

Remote origin

- `git remote add origin <url>`



National Czech Programme | eosc

GitHub

GitHub

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Propojení (local – remote)

A) Máme lokální repozitář → **potřebujeme ho odevzdat (nahrát) do GitHubu**

- `git remote add origin <url>`
  - `git fetch origin`
  - `git branch --set-upstream-to=origin/<branch> main`

Propojení (local – remote)

B) Nemáme → **stáhneme ho z GitHubu**

```
git clone <url>
```

Vytvoření repozitáře

1. **Registarce**
2. **Vytvoření repozitáře (projektu)**

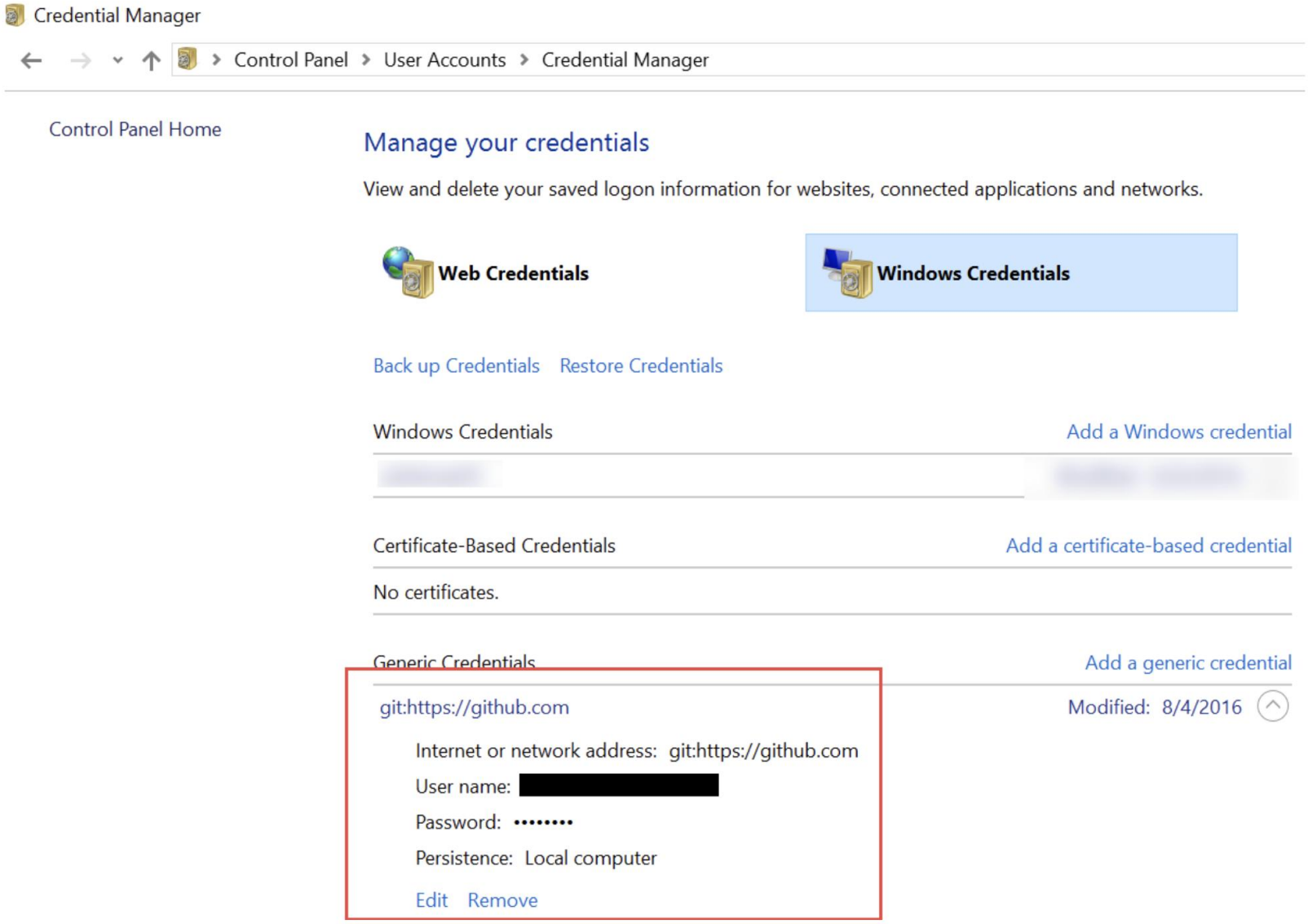
Propojení (local – remote)

Jak propojit lokální Git s GitHub?

A) SSH => `ssh-keygen -t ed25519 -C "pacesm@tf.czu.cz"`

B) **HTTPS** => Token (PAT)

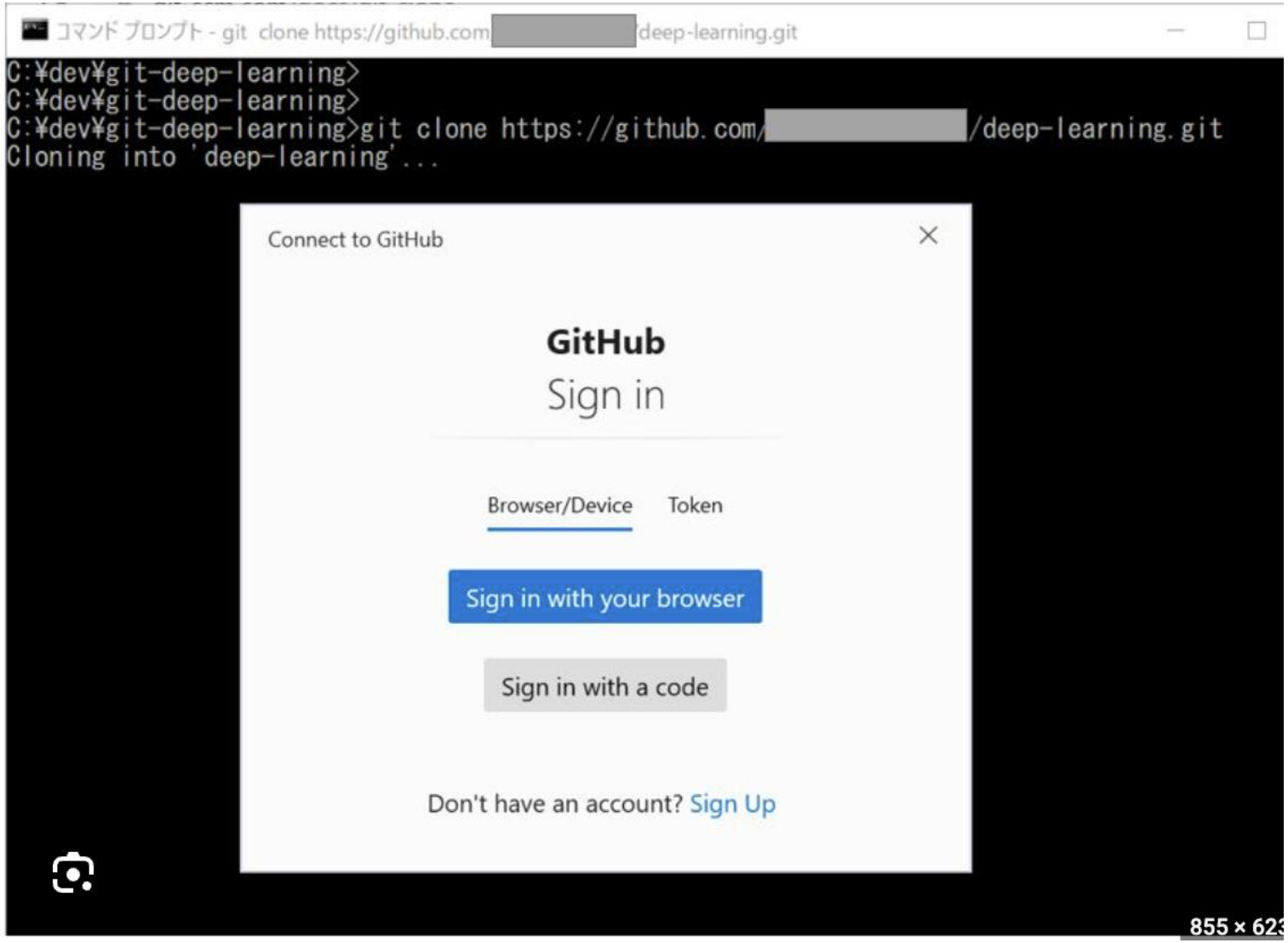
```
marek@Mac dummy-one % git push
remote: Permission to eosctest/dummy-one.git denied to pacesmarek.
fatal: unable to access 'https://github.com/eosctest/dummy-one.git/': The requested URL returned error: 403
marek@Mac dummy-one % git push
Username for 'https://github.com': eosctest
Password for 'https://eosctest@github.com':
Everything up-to-date
```



Windows Credential Manager – Correct Github credentials

https://snede.net/git-does-not-remember-username-password/

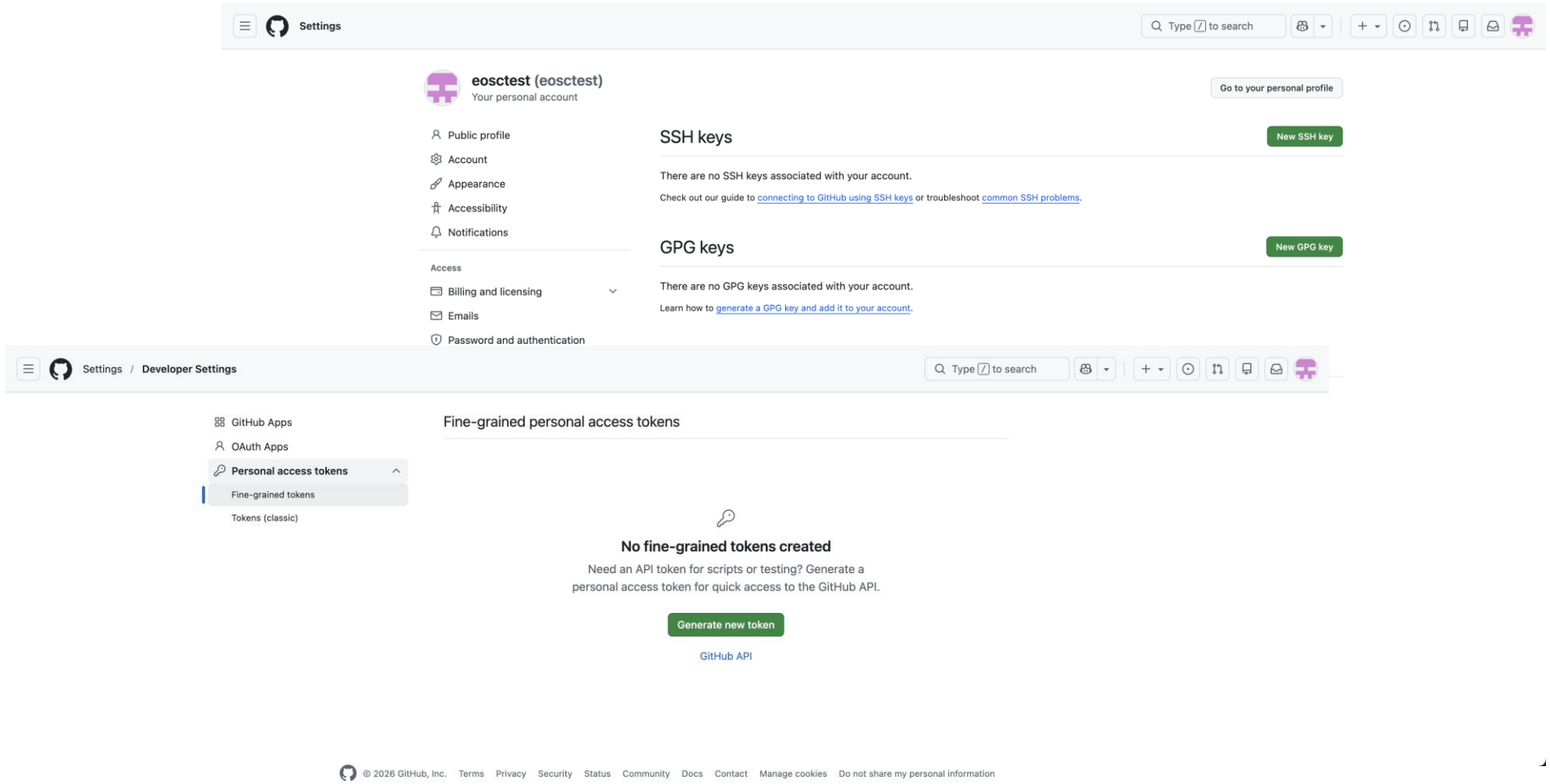
B) HTTPS => Token (PAT)



https://rainbow-engine.com/github-howto-use/

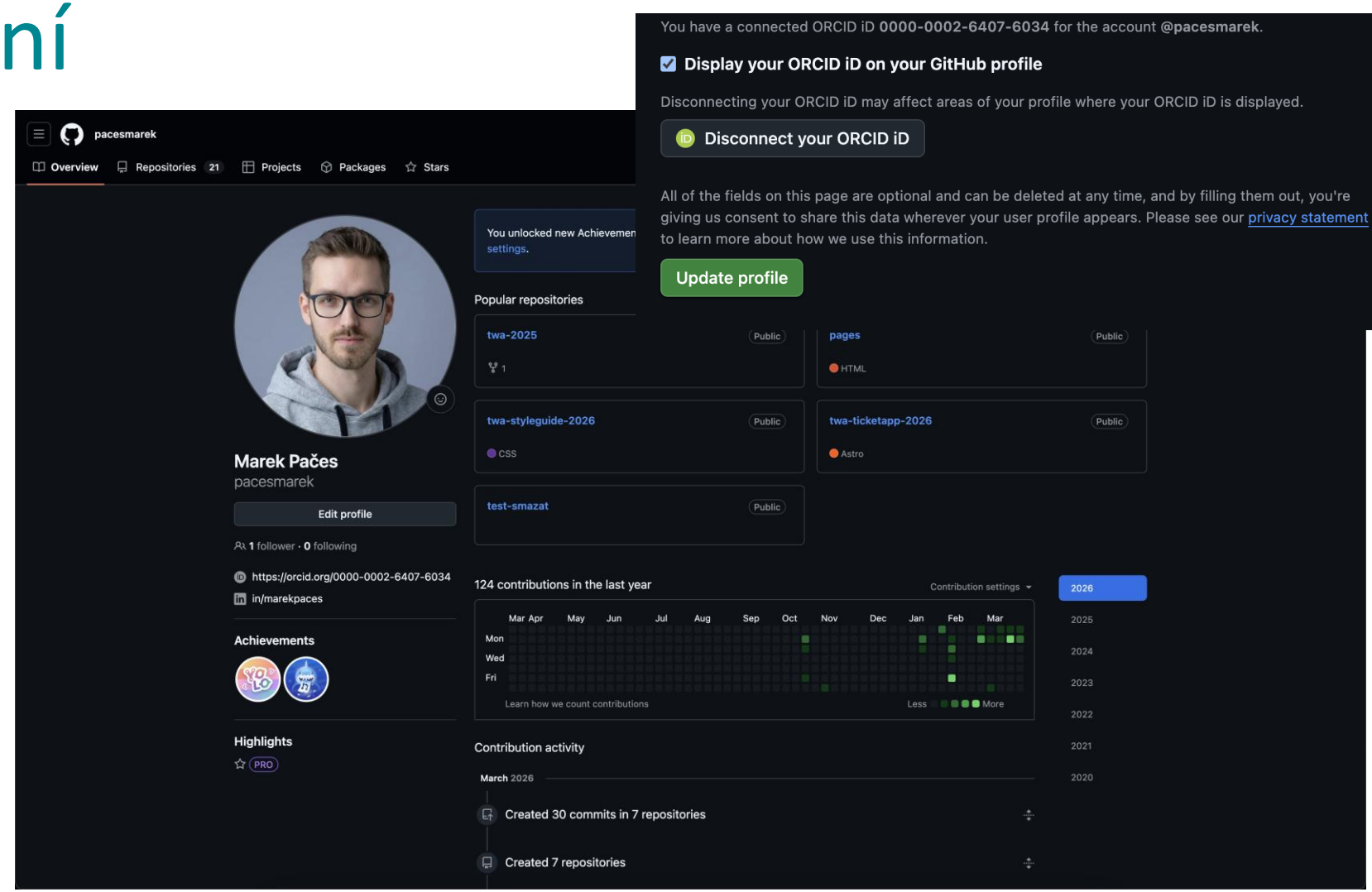
Generování tokenu / Zadání SSH

- SSH
- PAT



Úvodní nastavení

- Profile
  - ORCID
- Copilot



Hlavní oblasti

1) Repositories

2) Issues + Branches

3) Pull/Merge request

4) Projects

- Codespaces
- Tags (Releases)
  - `git tag -a v1.0.0 -m "Verze 1.0.0 - stabilní release“`
  - `git push origin v1.0.0`
- Pages
- Actions (pipelines, CI/CD)

National Czech Programme | eosc

LinkedIn

Děkuji za pozornost

mailto: pacesm@tf.czu.cz

a href: www.marekpaces.cz

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY